IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 24, 2019, accepted November 2, 2019, date of publication November 12, 2019, date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953086

IES-Backbone: An Interactive Edge Selection Based Backbone Method for Small World Network Visualization

CHENG ZHAN[1,2,3], DAOBING ZHANG[1,2], YANG WANG[1,2], DAOYU LIN[1,2], AND HUI WANG[1,2,3]

[1]Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[2]Key Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[3]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Daobing Zhang (dbzhang@mail.ie.ac.cn)

ABSTRACT Visualization of the small world network is an excellent challenge for classic layout algorithm, which is highly connected, resulting in the shape of the hairball. Backbone extraction method can simplify the classic layout to get better visualization, and it has become a significant approach in this field. However, contemporary approaches have two primary defects. Centrality based method loses plenty of topology information, and most contemporary approaches have the problem in low interactivity due to parameter sensibility. We proposed a backbone method based on interactive edge selection (IES-Backbone) to solve two problems for small world network visualization that mentioned above. The proposed method starts with backbone extraction of the network and then apply the layout algorithm to get visualization results. A critical approach of the backbone method is edge selection, which is based on the distance between vertices layout of the binary stress model. Edge selection makes the simplified network a clear community structure feature with more topological details. The simplified network is high in homophily and has closer average path length to the original network. The visualization result is controlled by edge limit ratio r and sampling rate s. Different choices of two parameters can change the results substantially on visual without affecting the layout quality, which proves high interactivity for users. Experiments prove that IES-Backbone is an interactive visualization method that presents community and sufficient topological features.

INDEX TERMS Visualization, graph layout, small world network, backbone.

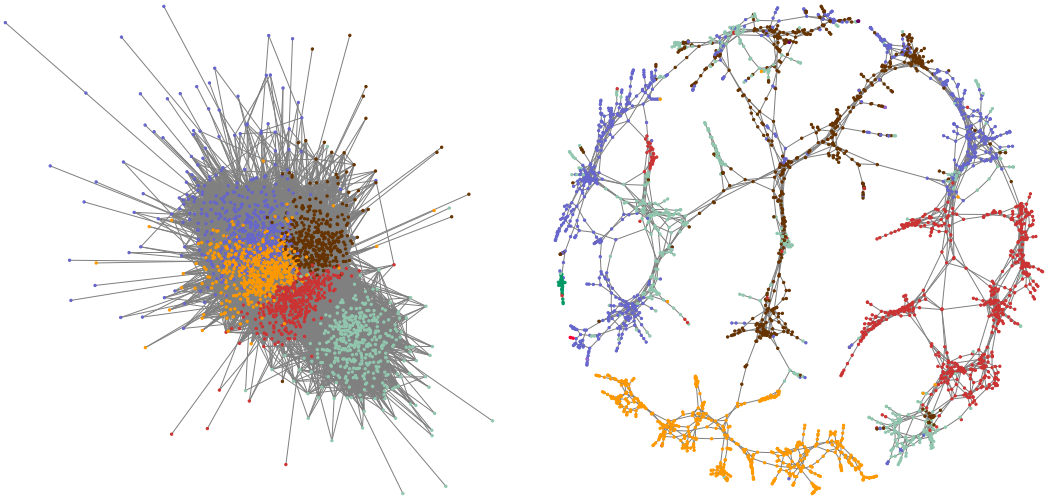
I. INTRODUCTION

An essential feature of social networks is that the nodes in networks are highly connected, and most of the nodes are connected to each other. Large-scale complex social networks have a large number of nodes, while the shortest distance between vertices is small, and any two nodes are accessible within a small number of hops. The complex social networks with such features are often referred to as small world networks [1].

People hope to solve many sociological problems with the method of network visualization [2]–[4]. Small world network visualizationis a challenging task and a vital topic in social network analysis [5]. Although there are plenty of well-designed force-directed methods to the layout at present, users cannot receive a satisfying result when applying existing layout algorithms because of the complication in the node-link relationship. The example of Fig.1(a) shows that the result displays in the shape of a hairball. It cannot reveal the inherent structural features of the network, either can people visually obtain any useful information. Therefore, it is a visualization of reduced readability.

It is necessary to simplify the complex connection of networks so it will not be constrained by too many nodes and edges during layout, allowing a more distinct small world network. The proxy graph [6] presents a feasible simplification approach. The rationale is to create a proxy graph whose size is much smaller to replace the original graph. The proxy graph makes it easier for people to analyze the original graph, such as visualization. Various techniques can generate proxy graphs, and graph filtering [7]–[9] is one of the most common. Graph filtering gives weights to edges or nodes, and

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.



(a) (b)

FIGURE 1. The comparison of hairball and backbone. (a) Visualization result using the graph drawing algorithm directly. It displays a shape of hairball because the original network was not processed. (b) Visualization result after backbone extraction, showing a clear structure to users.

filter the original graph to obtain a subgraph. The subgraph is the *backbone* that is used to a layout.

There are two main problems with the current backbone method. First, the method causes a massive loss of topological information in the network, although the simplified graph is highly readable and shows the connections of vertices well. This shortcomming will be shown in our experiments. Second, as the visualization quality is sensitive to the parameters, such as sampling rate, it is difficult to find an appropriate value to obtain a satisfactory layout result [10]. In detail, centrality based backbone method lost too much topological information because they simply used a tree to organize the backbone, which overly simplified the backbone. Moreover, the incorporated filtering parameters of quadrilateral simmelian backbone method have to be selected manually and individually for each input instance. This feature reduces the interactivity of the algorithms.

Using the centrality of a graph to visualize small world network can yield a good result. Removing high betweenness edges will result in a structurally meaningful abstraction. We found that most of the same community nodes are in the same branches of the tree, achieving the simplified network with high homophily, which is a terrific attribute in small world network visualization. However, doing so will still lose the intrinsic topology in the graph.

We present a small world network visualization technique called IES-Backbone that can maintain good readability in the layout while preserving the internal topology of the network. Fig.1(b) is the visualization result of our method. Compared to the previous works [8], [9], we designed a new edge selection strategy based on the distance between vertices to obtain backbone, and used bStress model to make this backbone method more efficient. Our main contributions are following:

1. We proposed an interactive backbone method based on an edge selection approach to avoid hairball layout of small world network visualization. By controlling the edge length limit r and sampling rate s, users can easily change the visualization result without affecting the layout quality.

2. Instead of using classic force-directed model, we chose to use a more efficient binary stress model layout method in graph drawing to make the visualization result more readable.

3. The IES-Backbone method not only maintains topological details of the network but also reflects the community attributes of nodes. The high homophily of the simplified network also proves that this approach allows the network to suggest good community features.

II. RELATED WORK

There are two primary tasks in small world network visualization: backbone method and graph layout. Backbone method chooses the most important edges from the original network to simplify small world networks, and the graph layout algorithm largely determines the visualization quality. In addition, many evaluation criteria had been proposed to judge visualization quality.

A. BACKBONE METHOD

Small world networks have a large number of edges, and users can get a simplified network by performing edge filtering while maintaining the number of nodes. The network is called the backbone after filtering. There are usually two steps in obtaining a backbone network. The first step is the edge embedding. It assigns different importance to each edge. The second step is edge filtering. It chooses which edge to use as the backbone based on step one and all of the nodes will be serving as the backbone. Edge filtering also affects the final layout. The method of edge filtering emphasizes the demonstration of the graph features, including retention of

the structural features such as graph connectivity or cuts [11], spectra [12], [13], distance preserving [14], [15], etc. Three typical backbone methods are shown below.

According to Simmel's sociological thinking [16], Nick *et al.* [7] proposed the Simmelian backbone method. They initialized the edge weights at first by using some embedding criteria, such as the number of triangles contained in edges. The edges were then re-weighted by comparing the ordered neighborhoods of its two vertices to achieve the final edge embedding process. The method removes the relatively unimportant edges by setting a threshold of edge weights. The extracted backbone was combined with the left edges and all the nodes in the origin network. This method effectively organizes nodes of the same cluster. Most of the remaining edges are connected to nodes of the same community, however, they do not maintain the connectivity of the graph.

Based on the Simmelian backbone method, Nocajet *et al.* [8] proposed an improved method called quadrilateral simmelian backbone which generated a maximum spanning tree to connect the nodes, ensured that all nodes were connected, and used sampling rate to extract other edges. Also, they proposed a new criterion for edge embedding based on the weighted accumulation of triangle in quadrangles. By comparing to the utilization of the Jaccard coefficient and density in edge embedding, they concluded that applying this method with the maximum spanning tree could yield better results. The improved method is still slightly inadequate [10]: threshold setting greatly affected network visualization, and it still needs to manually adjust the parameters to optimize the effect of the backbone extraction. Seeing the problem, they designed a method to choose the threshold [10] adaptively.

Van Ham and Wattenberg [9] proposed using the graph centrality indicators to give edge weights and the minimum spanning tree to establish a simplified network as the backbone. The extent which nodes or edges in the network are close to the center is called centrality, and it represents how important they are in the network. Nodes or edges with influential centrality act as a ''bridges'' in the network, and they are usually located on paths connecting two different clusters. This method can clearly express the structural results of the visualization. However, it loses part of the topology from origin network, and there are no edge compensation mechanisms in user interaction, which ultimately leads to the inability to see a detailed network structure. Edge *et al.* [17] also used the betweenness centrality in the network as a edge filtering strategy in thier work.

We have developed our new method by combining the graph centrality indicators mentioned in Van Ham's method [9]. We also compared the quadrilateral simmelian backbone [8] method with our IES-Backbone.

B. GRAPH LAYOUT ALGORITHM

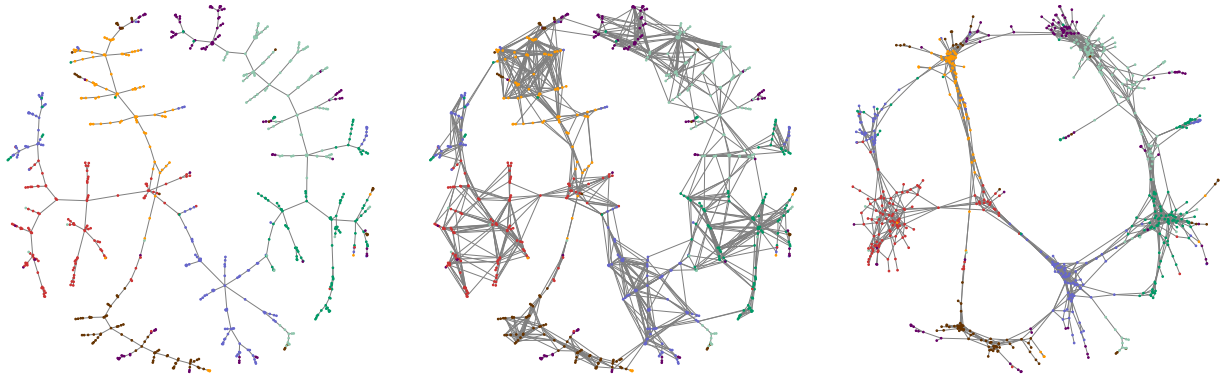
Drawing a graph is of more efficiency in understanding network relationships than only viewing data [18]. Numerous scholars have been attracted to study graph drawing due to its high readability and intuitive visualization of the node-link diagram. Research on the graph drawing is known for its force-directed model, and various versions have been developed to improve the layout quality.

Force-directed model follows the simulation of a spring system, in which the length of the spring is proportional to the force exerted by an extended spring [19]. The first force-directed model was originally proposed by Eades [20]. He built a rigorous physics system that imagined the nodes as rigid rings and the sides as springs. Kamada and Kawai later proposed the KK model [21]. The model discussed the concept of the ideal distance in the layout and defined the total energy equation of the spring system. Fruchterman and Reigold [22] also improved the models of Eades model. They introduced the repulsive force between the nodes and fixed the node coverage. This model, also known as the FR model or the spring model, is the default layout algorithm for many visualization software. The main problems of such early models include high-level of computational complexity, which is not conducive to the layout of large-scale networks, and the possibility of falling into local optimum. Some refined algorithms have been proposed to improve the force-directed layout model. As an example, ForceAtlas2 was proposed by Jacomy *et al.* [23] to obtain a layout for network more quickly and precisely. This algorithm extends from FR model.

When De Leeuw and Michailidis [24] studied the KK model, he found the model had the same mathematical expression as the stress model in the field of multidimensional scaling analysis. Later, Gansner *et al.* [25] used the stress majorization to figure out the KK model and optimized its final layout quality. Inspired by this advancement, some layout algorithms based on stress model were proposed. Koren and Çivril [26] proposed a binary stress model (bStress), which aimed to arrange all the nodes, being as close as possible, within a circle evenly. Also, the author suggested using the Barnes-Hut structure in stress majorization to deal with the Euclidean problem. The steps of the distance are accelerated, and the efficiency of the method is operational.

C. EVALUATION CRITERIA

A feature of graph layout is the match between structural adjacency and graphical proximity [27]. However, the layout algorithm often fails to depict the complex relationships faithfully [28]. Typical indicators of visualization quality (traditionally known as aesthetics in the field of graph layout) are the number of edge crossings, the angular resolution at vertices, the alignment of paths with straight lines connecting their origin and destination, and more [29]. In addition, graph layout algorithms are typically based on optimization of layout objectives that can be interpreted as quality criteria [27]. These are some standard criteria for graph layout algorithms; however, they do not provide a proper assessment of specific visualization task.



(a) MST layout result (b) edge selection (c) final layout result

FIGURE 2. Edge selection process and secondary layout on Caltech36 with $r = 0.15$ and $s = 0.25$. Coloring nodes based on the results of Louvain method and coloring does not enter the calculation of the layout. (a) Layout result of minimum spanning tree. (b) Layout result after interactive edge selection. Most of the added edges connect vertices from the same cluster. (c) The final layout of the network.

There are two approaches in the evaluation of small world network visualization. One is to evaluate them through quantitative metrics, and the other is to judge the visualization results visually. In terms of quantitative metrics, homophily of a graph is used to denote the ratio of intra-cluster edges [7], [8]. The higher the value, the better the visualization results represent the community structure feature of the network. The average path length and clustering coefficient can describe the structural feature of small world networks [30], [31] well. On the other hand, users can observe the overlap of nodes between communities to judge the visualization effect directly [17]. The two metrics are also used to evaluate the visualization result of small world network [8], [10].

III. IES-BACKBONE METHOD

In this section, we will describe and explaine each key step of our method at first and the flow of the algorithm will be present at the end of this section. Fig.2 shows the process of our method.

A. EDGE FILTERING BASED ON CENTRALITY

In graph theory and network analysis, indicators of centrality identify the most important vertices or edges within a graph. Betweenness centrality quantifies the number of times a node or an edge acts as a bridge along the shortest path between two other nodes. For small world networks, nodes or edges with high betweenness centrality mean that they connect to different communities, and the edges connecting different clusters have less centrality [9]. Therefore, this indicator is going to be used to make edge filtering.

Centrality indicators identify the most critical vertices or edges within a graph in graph theory and network analysis. Betweenness centrality quantifies the number of times a node, or an edge acts as a bridge along the shortest path between two other nodes. For small world networks, nodes or edges with high betweenness centrality mean that they connect to different communities, and the edges connecting the same cluster have less centrality [9]. Therefore, this indicator can be used for edge filtering.

For an undirected graph graph $G = (V, E)$, the betweenness of a vertex $v \in V$ in a graph is computed as follows [32]:

$$C_v(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{1}$$

where σ_{st} is the number of the shortest paths from s to t and $\sigma_{st}(v)$ is the number of those paths that contain vertex v. Similarly, the betweenness centrality of an edge e is defined as the frequency at which one of the shortest paths between two vertices occurs:

$$C_e(e) = \sum_{s \neq v \neq t \in V} \frac{\rho_{st}(e)}{\sigma_{st}}, \tag{2}$$

where $\rho_{st}(e)$ is the number of the shortest path from s to t including edge e.

We assumed the input network was unweighted and connected After calculating the betweenness centrality of all edges in the network, we weighted each edge to obtain a weighted graph that marked the importance of all edges. We performed edge filtering by generating a minimum spanning tree, which enabled vertices of the same cluster to be in adjacent branches. This minimum spanning tree is called Tree $= (V, E_{\mathrm{MST}})$. As the edge number of a tree is one less than the node number, we minimized the size of the small world network, and it was the smallest graph that maintained its connectivity. The tree structure that was used to process the graph can effectively reduce the complexity of a small world network.

B. BINARY STRESS MODEL

Edge filtering simplifies the complexity of the network. We used the binary stress model [26] to create a more readable network layout. The stress function of bStress is:

$$B(p) = \sum_{e(i,j) \in E} \left\| p_i - p_j \right\|^2 + \alpha \sum_{i \neq j \in V} (\left\| p_i - p_j \right\| - 1)^2, \tag{3}$$

where $e(i, j)$ is denoted as an edge connecting vertex i and j, p_i is the position of vertex i. The first part relates the layout to the graph structure ensuring that edges are short, and the second part makes the nodes evenly distributed within the circle. α is a parameter that balances two terms. The stress function can be solved by stress majorization [25]. The functions in (4) were derived from (3) and made it equal to zero. They are used to solve the problem iteratively.

$$(M + \alpha L)x(t+1) = b^{x(t)}, (M + \alpha L)y(t+1) = b^{y(t)} \quad (4)$$

When calculating the derivation of the stress function, there are two $|V| \times |V|$ matrices defined as L and M. They can be constructed according to the structure of the input network. In addition, $b^{x(t)}$ and $b^{y(t)}$ are two vectors determined by nodes' position in the last iteration. Then the stress majorization process can be solved by using the conjugate gradient method.

$$L_{ij} = \begin{cases} -1 & e(i,j) \in E \\ \sum_{k \neq i} L_{ik} & i = j \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$M_{ij} = \begin{cases} -1 & i \neq j \\ |V| - 1 & i = j \end{cases} \quad (6)$$

$$b_i^x = \sum_{j \neq i} \frac{x_i - x_j}{\|(x_i, y_i) - (x_j, y_j)\|} \quad (7)$$

$$b_i^y = \sum_{j \neq i} \frac{y_i - y_j}{\|(x_i, y_i) - (x_j, y_j)\|} \quad (8)$$

This layout idea provides good initial conditions for edge selection. The first layout result P_{first} is showed in Fig.2(a) where we use Caltech36 as an example.

C. INTERACTIVE EDGE SELECTION

Although the layout result of the minimum spanning tree was visually decent, the structure of the network was still too simple with many edges being filtered. It was difficult to capture the inherent structural features of the network. Obviously, more edges need to be reinserted into the network so that users can get more useful information from the visualization result. Our goal was to build a complex intra-cluster structure of the network, which reflects more topological relationships. We have made a good vertices position distribution in the step of drawing the minimum spanning tree and placed vertices from the same clusters in a closed position.

Next, we calculated the Euclidean length of all edges in the original graph under the current layout.

$$\text{dist}(e(u, v)) = \sqrt{(p_u^x - p_v^x)^2 + (p_u^y - p_v^y)^2} \quad (9)$$

After applying bStress model to the MST, we considered that the lengths of edges connecting vertices of the same clusters were small while the edges connecting different clusters of nodes were relatively long(later experiments can support this). Therefore, if we choose a good upper limit ratio of edge length $r \in [0, 1]$, we can get an edge set E_{backup} where most of edges connect the same cluster nodes.

The sampling rate $s \in [0, 1]$ controls the number of reinserted edges. A certain number of edges were randomly selected from E_{backup} based on s to determine the complexity of the final network. More edges were selected because of the high sampling rate, as a result, providing more topology information for the final network. The set of selected edges is called E_{select}. Figure 2(b) shows the effect of inserting edges with $r = 0.15$ and $s = 0.25$.

In addition, the result of EBC and MST of a network are identical, and we can save time by calculating EBC and MST once when testing parameters.

After edge filtering and edge selection, the final network $G' = (V, E_{\text{MST}} \cup E_{\text{select}})$ contains the backbone of the original network and more details about topological structure. Through the secondary layout, we will get the result of the network visualization $P \in R^{|V| \times 2}$.

The position of each vertex already had an initial coordinate in MST layout. These positions had been roughly determined and did not require much adjustment. As the added edges were mainly used to connect vertices from the same cluster, the structure of the network did not change significantly. A second layout can be done with fewer iteration. Fig.2(c) shows the final result of the visualization algorithm. When running the algorithm, the first layout can be iterated up to 300 times, while the second layout can be iterated up to 50 times to save time for adjusting parameters.

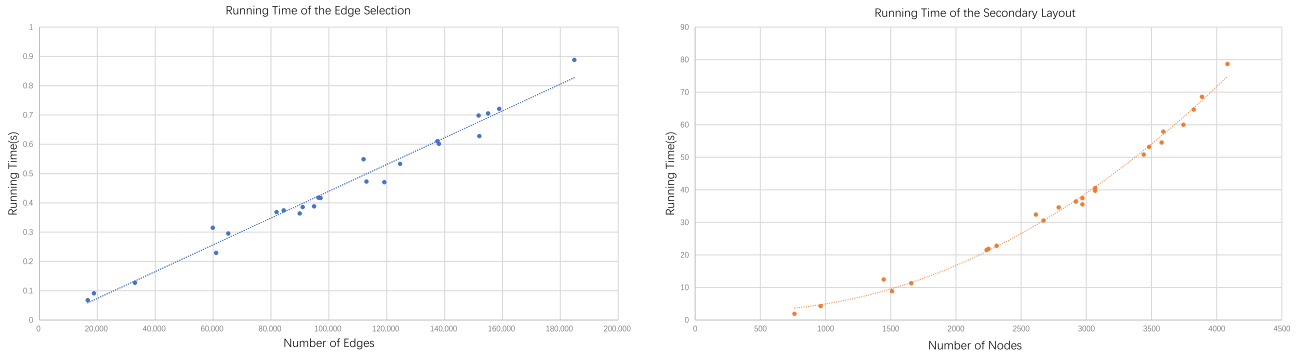
The centrality based method can cluster the same class nodes very well, but the simplified network is too simple because they used a tree to organize the backbone, which overly simplified the backbone. So we propose our interactive edge selection approach to restore the topology details of the network. At the same time, the method relies on an efficient layout algorithm. After experimentation, we obtained the best results using the bStress model. These two main contributions are at the heart of our approach.

Finally, we summarized the working flow of IES Backbone, as shown in alg1. The process begins with an input of a small world network G and two parameters r and s. The algorithm returns the position of each vertex P on the screen. The process is as follows: first, we conducted edge filtering based on edge betweenness centrality and got a minimum spanning tree. Next, we applied the bStress model to obtain the layout of the MST. Then, performed the interactive edge selection to get the backbone of the network, and finally performed the secondary layout to get the result.

D. RUNNING TIME

There are four steps in the algorithm: EBC calculating and MST generation, first layout, edge selection, secondary layout. The following is an analysis of the running time of the four steps.

It is necessary to calculate the shortest path between all pairs of when calculating EBC, which is a time-consuming process, costing in $O(|V|^3)$. Brandes [33] proposed a more



(a) (b)

FIGURE 3. (a) The running time of the edge selection. (b) The second layout step with the data of Facebook100.

Algorithm 1 IES-Bacbone Method

Input: original network $G = (V, E)$; upper limit ratio of edge length $r \in [0, 1]$; edge sampling rate $s \in [0, 1]$

Output: Position of every vertex $P \in R^{|V|\times 2}$

1: EBC(e) $\leftarrow$ betweenness centrality of every edge in G
2: Tree $= (V, E_{\text{MST}})$ $\leftarrow$ generate a minimum spanning tree from G with EBC(e)
3: P_{first} $\leftarrow$ compute the layout of the Tree with bStress model
4: **for** each $e \in (E - E_{\text{MST}})$ **do**
5: dist(e) $\leftarrow \|P_{\text{first}}(e.\text{source}) - P_{\text{first}}(e.\text{target})\|$
6: **end for**
7: $E_{\text{backup}} \leftarrow \{e : e \in (E - E_{\text{MST}}) \wedge \text{dist}(e) \leq r \times \max(\text{dist}(e))\}$
8: $E_{\text{select}} \leftarrow$ Randomly select $s|E_{\text{backup}}|$ edges from E_{backup}
9: $P \leftarrow$ layout the final graph $G' = (V, E_{\text{MST}} \cup E_{\text{select}})$ by bStress model with P_{first} as the initial position

efficient algorithm that for betweenness centrality, and EBC can be computed in $O(|V||E|)$. Time complexity of MST generation in this algorithm is smaller and it can also be generated only in $O(|E|\log|V|)$.

Solving bStress layout involves two steps: calculating intermediate variables and solving the conjugate gradient, which is $O(|V|\log|V|)$ and $O(|V|+|E|)$ [26] for one iteration. It takes several iterations to get a good drawing.

In edge selection, each edge needs to be traversed twice. The first traversal computes the length of all edges, and the second traversal selects edges randomly. Therefore, the complexity in this step is $O(|E|)$. The algorithm of the second layout is completely identical to the first layout, so the time complexity is the same. However, the secondary layout does not need many iterations, so the running time will be short. In our experiments, the secondary layout only spends one-sixth time compared to the first layout.

Because the layout process is the most time-consuming step, the running time of the algorithm depends mainly on the layout algorithm. The time complexity of our method is $O(|V|\log|V|)$. In other words, our edge selection operation does not consume too much time. Fig.3(a) and Fig.3(b) show the running time of the edge selection step and the secondary layout step. The trend of the running time is in line with our analysis of the time complexity.

IV. EVALUATION AND RESULTS

A. IMPLEMENT AND DATASET

We implemented the framework of visualization algorithm in Python3 and also used NetworkX [34] to compute EBC and generate the minimum spanning tree. The following experiments were all run on an Intel Core i7-7700HQ computer with 16 GB of RAM. We have uploaded the code of our IES-Backbone at https://github.com/tomzhch/IES-Backbone.

We used the Facebook100 dataset [35] as real-world data in our experiments. This dataset contains friendships of Facebook users from 100 colleges in the United States. The size of the 100 networks varies from 762 to 41K, and the number of edges ranges from 16K to 1.6M. The dataset includes various attributes of the node as well, such as gender, major, dormitory, etc. Among attributes, ''dormitory'' is considered to be important for the creation of social relationships in many networks. However, it does not provide ground-truth group structure. We planned to apply a community detection algorithm which is Louvain method [36] to the network to get an approximate ground-truth. The ''dormitory'' attribute and community from Louvain method would be used to evaluate our algorithm.

B. QUALITY METRICS

One of the causes of the low readability of a layout result is visual clutter between communities [17]. A good backbone should have more inter-cluster edges and less intra-cluster edges. The ratio of homophily edges can illustrate how good a backbone is. Therefore, we used homophily of a graph as the quality metric.

$$\text{homophily}(G) = \frac{\#\text{ho\_edges}}{\#\text{ho\_edges} + \#\text{he\_edges}} \tag{10}$$

#ho_edges is the number of homophily edges and #he_edges is the number of heterophily edges. Homophily edges are the edges whose source and target vertices are from the same cluster, while heterophily edges are the opposite. For vertices that lack attributes, the edges to which they are connected are ignored when testing quality metrics of a graph using the ''dormitory'' attribute.

Our primary focus in this algorithm is to increase the ratio of homophily edges and suppress the number of heterophily edges. So, we need a metric to measure the change of homophily after our interactive edge selection. We hope to find the combination of parameters that maximize the addition of homophily edges based on the following metric.

$$\text{diff\_homophily}(G) = \frac{\#\text{ho\_select}}{\#\text{ho\_select} + \#\text{he\_select}} \tag{11}$$

#ho_select and #he_select are the number of homophily edges and heterophily edges in edge selection. Because the final network is $G' = (V, E_{\text{MST}} \cup E_{\text{select}})$, only the edges in E_{select} are included in the calculation of diff_homophily(G). In other words, diff_homophily(G) measures how many homophily edges are selected in the step of interactive edge selection. In summary, homophily(G) suggests the quality of the visualization algorithm and diff_homophily(G) shows how much the edge selection contributes.

Homophily is a metric of the ability of an algorithms to suggest network community feature. Compared to it, the major metric that can describe the structure feature of the small world network is average path length (APL).

$$\text{APL}(G) = \sum_{s \neq t \in V} \frac{d(s,t)}{n(n-1)} \tag{12}$$

$d(s, t)$ is the shortest path length from s to t and n is denoted as the node number of G. The nodes in small world networks are highly connected with very few hops. Therefore, the average shortest path length of such a network is only 2 to 3. Reducing the complexity of the network allows for excellent visualization, but it inevitably loses topology details. Therefore, it is necessary to maintain as much topology as possible when getting the backbone. The change in the average shortest path length indicates the performance of different backbone methods in maintaining topology details. The closer the APL value is to the original network, the stronger the ability to maintain topology details.

C. RESULTS AND DISCUSSION

1) RESULTS OF IES-BACKBONE

We applied the IES-Backbone method to Facebook100 dataset and compared the visualization results to the force-directed model layout without the backbone method. The force-directed layout model we used was the FR model [22] and the KK model [21]. The bStress model is used in IES-Backbone, and it is necessary to incorporate the model into the comparison to demonstrate the necessity of the backbone method. We also tested the result of centrality based visualization method [9].

At this part, we evaluated our method qualitatively by comparing the results on visual. Fig.5 displays the comparison. As shown, the classic layout algorithm such as FR and KK model cannot process the small world networks and result in the shape of a hairball. The result of centrality based method [9] is notably better than the classic layout algorithm. The centrality based method simplified the network into a tree, and the layout result shows a clear structure. However, it was too simple to suggest the original structure of the small world network. We will discuss this problem in the following session. IES-Backbone method not only prevented the hairball layout but also maintained topology and community features of the network. The visual clutter of IES-Backbone is the least, and the nodes of each community are clearly distributed on the plane. Admittedly, the visualization used by IES-Backbone method obtained the best visual result.

We also tested our approach in synthetical generation networks using the Stochastic Block Model (SBM) [37]. This network has the basic facts of community attributes. As shown in Fig.4, it provided a proper layout with sufficient topology details after applying our IES-Backbone method. Also, nodes belonging to different communities were separated. Simplified network homophily reached 0.9308, which means it had adequately reflected the characteristics of the community network.

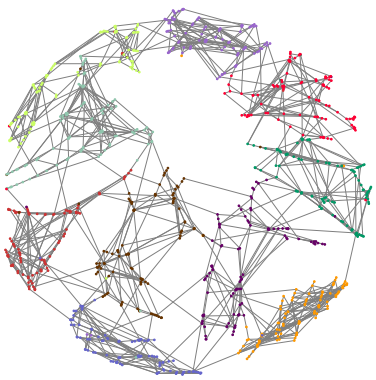


FIGURE 4. Visualization results of a synthetically generated network. The network has 1000 nodes, 9318 edges, and 10 communities. The visualization result provided an excellent layout with sufficient topology details after applying our IES-Backbone method. Also, nodes belonging to different communities were separated. Simplified network homophily reached 0.9308, which means it had adequately reflected the community characteristics of the network.

2) IMPROVEMENT ON TOPOLOGY DETAIL

Table.1 lists ten results of average path length (APL) in Facebook100 using different backbone method. The original network's APL is very small due to the property of the small world network. Centrality based backbone method [9] increase the APL considerably, indicating a massive loss of topology details and an overly simple visualization. Besides, quadrilateral simmelian backbone method [8] performs better than centrality based method, retaining more topology details in the visualization result.

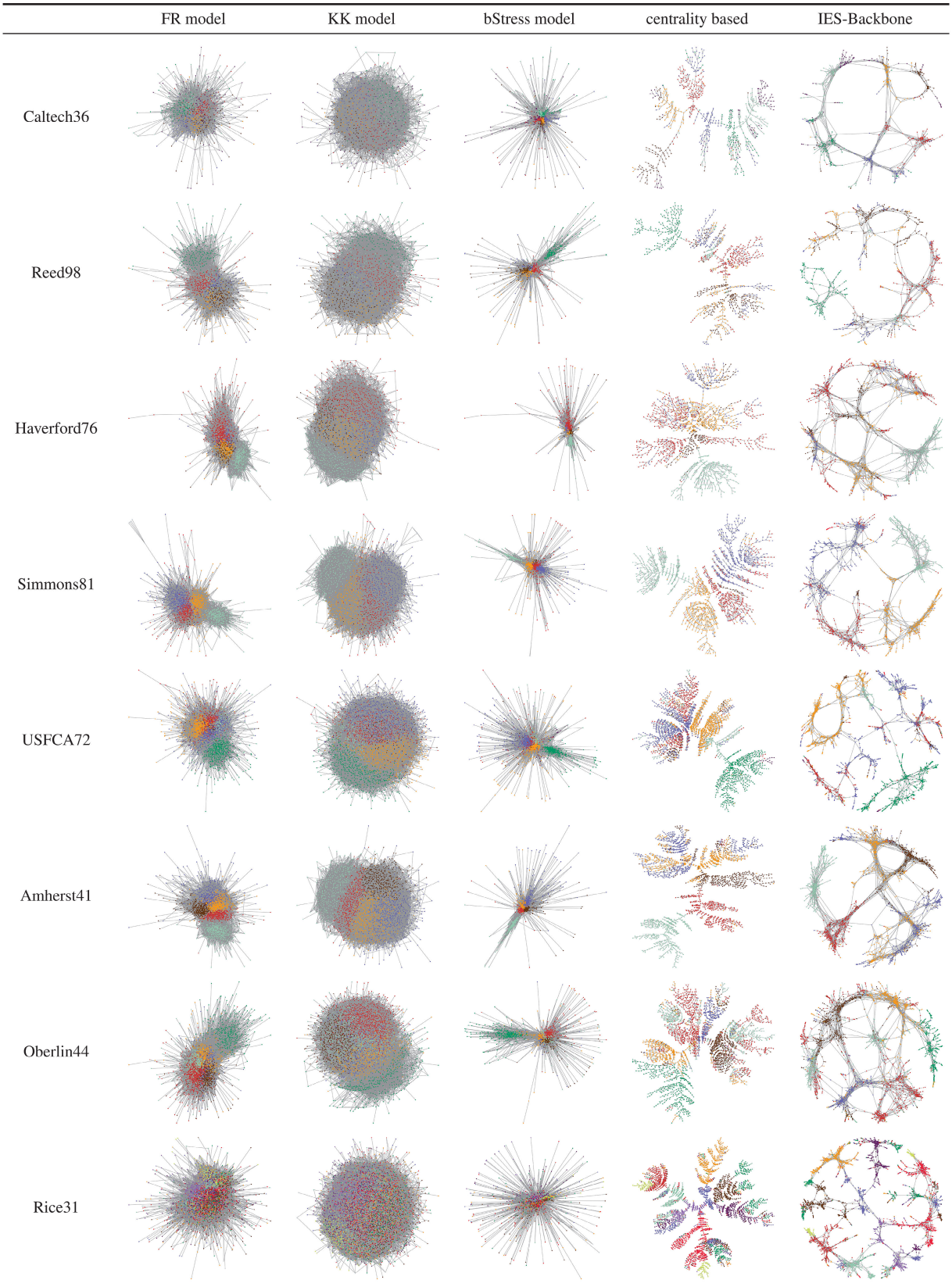


FIGURE 5. Visualization result in Facebook100.

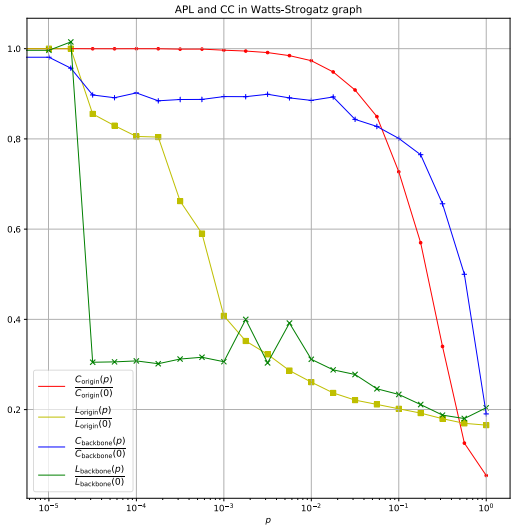


FIGURE 6. CC and APL curves in Watts-Strogatz graph. We plotted the curves of APL and CC in the original and simplified networks. Red and olive green lines represent the changes of CC and APL of original networks. The blue and green lines represent their changes in the simplified networks. The trend of the lines in simplified network is similar to that of the original networks. Therefore, the simplified network has maintained the features of the small world network.

TABLE 1. Average path length(APL).

Data Name	original network	Centrality based	Quadrilateral simmelian backbone	IES-Backbone (1)	IES-Backbone (2)
Caltech36	2.4	21.9	10.4	**9.7**	13.2
Reed98	2.4	22.7	**10.9**	12.4	16.6
Haverford76	2.2	20	8.2	**7.9**	11.8
Simmons81	2.5	19.7	13.4	**11**	14.6
Amherst41	2.4	32.4	10.7	**8.2**	13.4
Wellesley22	2.6	28.3	14.4	**9**	13.8
Oberlin44	2.5	25.9	14.3	**9.3**	12.8
Vassar85	2.5	33.1	14.9	**9.1**	14.9
Santa74	2.4	26	11.6	**8.3**	11.6

Regarding the IES-Backbone method, we chose two configuration parameters in the experiment. In Table.1, the parameter values of IES-Backbone (1) are $r = 0.15$, $s = 0.25$ and the value of IES-Backbone (2) are $r = 0.1$, $s = 0.35$. These two configurations approximate the edge numbers to the edge numbers in quadrilateral simmelian backbone method. The APL value in the simplified network is substantially less than the value in centrality based method and is generally less than the value in quadrilateral simmelian backbone method, which means our method can better perserve topological details.

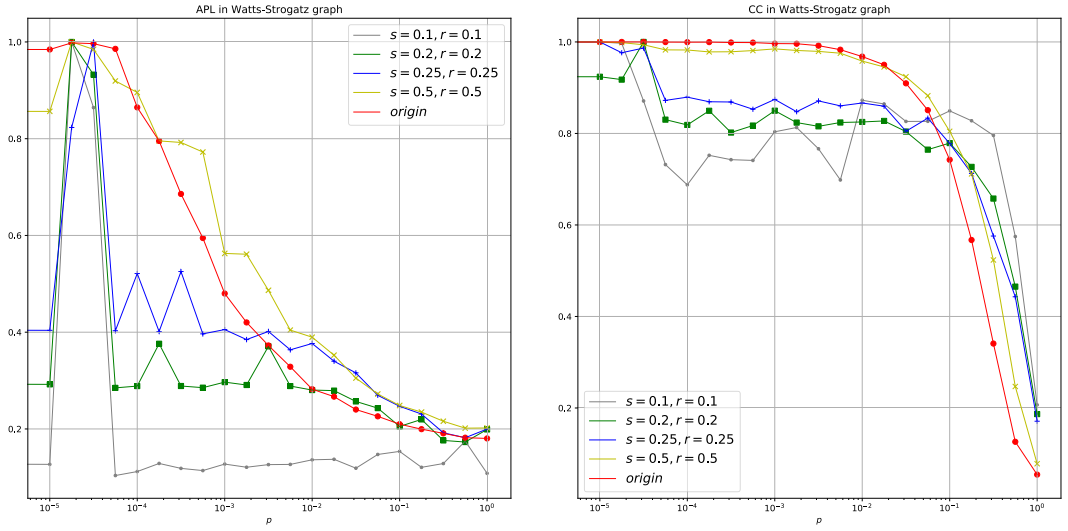
We also proved through experiments that the simplified network obtained by our method maintains the basic feature of small world networks. Small world networks are defined by two parameters, APL and clustering coefficient (CC). Watts-Strogatz model is a typical method for generating small-world networks [1]. In Watts-Strogatz model, during the transition from a regular network to a random network, APL begins to decline rapidly and then decelerates, while CC begins to change slowly and then declines rapidly. Such transition is a unique feature of the small world network. The curves of $\frac{\text{APL}(p)}{\text{APL}(0)}$ and $\frac{\text{CC}(p)}{\text{CC}(0)}$ express this rule [1]. p is a parameter in WS model, the probability of reconnection when constructing small world network. When $p = 0$, the result is a regular network, while when $p = 1$, the result is a random network. The change of p describes the process of WS small world network changing from regular network to random network. As shown in Fig.6, Red and olive green lines represent the changes of CC and APL of Watts-Strogatz networks. The blue and green lines represent their changes in the simplified networks. The trend of the lines in simplified network is similar to that of the original networks. Therefore, the simplified network has maintained the features of the small world network.

In addition, we tested the changes in APL and CC values from low to highly simplified networks. By controlling s and r, we choose four different simplified networks to compare with the original network. The result are shown in Fig.7(a) and Fig.7(b). As shown in Fig.7, when the original network is closer to the random network, our method can keep the small-world-ness properties well. On the contrary, when approaching the regular network, the ability to keep the small-world-ness properties is weakened, and the more simplified the network, the weaker the ability. This phenomenon is especially obvious in APL. When p is close to 0, the trend of APL and CC value in simplified network is not very similar to that of the original networks. As the simplified network becomes more complex, the trend of APL and CC value is closer to the original network.

3) EXPERIMENTS ON EDGE SELECTION

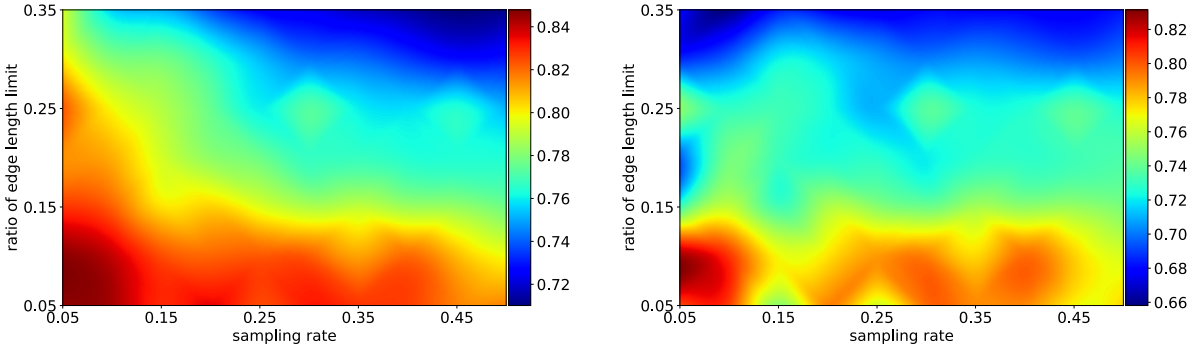
Two parameters affect the quality of the network visualization in edge selection: the upper limit ratio of edge length r and edge sampling rate s. When $r = 0$ or $s = 0$, the final network will be the simplest backbone, which is the MST we generate. Also, when $r = 1$ and $s = 1$, the final network is the same as the original network. The visualization layout is still a hairball if both r and s are 1, as shown in Fig.1(a). Fig.2(a) presents the result when $r = 0$ or $s = 0$, a layout of the MST, without more details of the network. Neither of these two configurations is ideal, and we need to find the best match in experiments.

Intensive parameter selection reduces the efficiency of the experiment, so we chose sparse parameters to test homophily of the graph and observed the trend. Take Reed98 as an example, as shown in Fig.8(a), the larger the r and s, the smaller the homophily of the final graph. Specifically, homophily is very sensitive to r. If r is too high, the layout result will deteriorate inevitably. The heatmap on homophily and layout figure shows that good results are concentrated in the area of $r \leq 0.35$. On the other hand, the effect of homophily on sampling rate is not evident as what it does to edge length.



(a) APL (b) CC

FIGURE 7. APL and CC curves in different simlified networks in Watts-Strogatz graph. (a) The curves of APL in the original and simplified networks. (b) The curves of CC in the original and simplified networks. Red lines represent the original networks. The other lines are the different degrees of simplified networks. When the original network is closer to the random network, our method can keep the small-world-ness properties well. On the contrary, when approaching the regular network, the ability to keep the small-world-ness properties is weakened, and the more simplified the network, the weaker the ability. This phenomenon is especially obvious in APL.



(a) homophily (b) diff_homophily

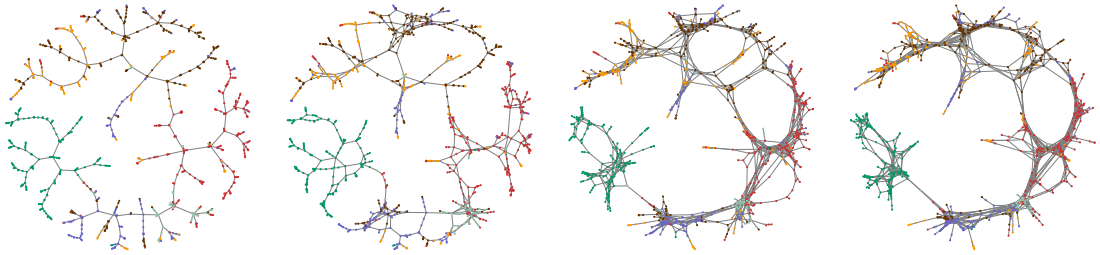
FIGURE 8. Trend in homophily and diff_homophily on Reed98. We used the heatmap to display the metric value under *r* and *s*. Red area indicates a high value, and the blue area is the opposite. Diff_homophily is meaningless when *r* or *s* is 1, so the heatmaps show the result starting at 0.05.

The trend of homophily also decreases with the increase of s, and experiments with other data yielded similar results.

The next step is to evaluate the effectiveness of edge selection. According to the result in testing homophily metric, network visualization quality was high in the area of $r \leq 0.35$ and $s \leq 0.5$. We conducted more experiments on diff_homophily. As seen in Fig.8(b), high value results are concentrated in areas where r and s are small, especially r. Although the highest metric of one network is usually in the lower-left corner of the heatmap, the complexity of the graph does not increase too much. It can be concluded that it is most effective to set a meager side length limit if users want to maximize the homophily. The short length edges are almost connecting the same cluster.

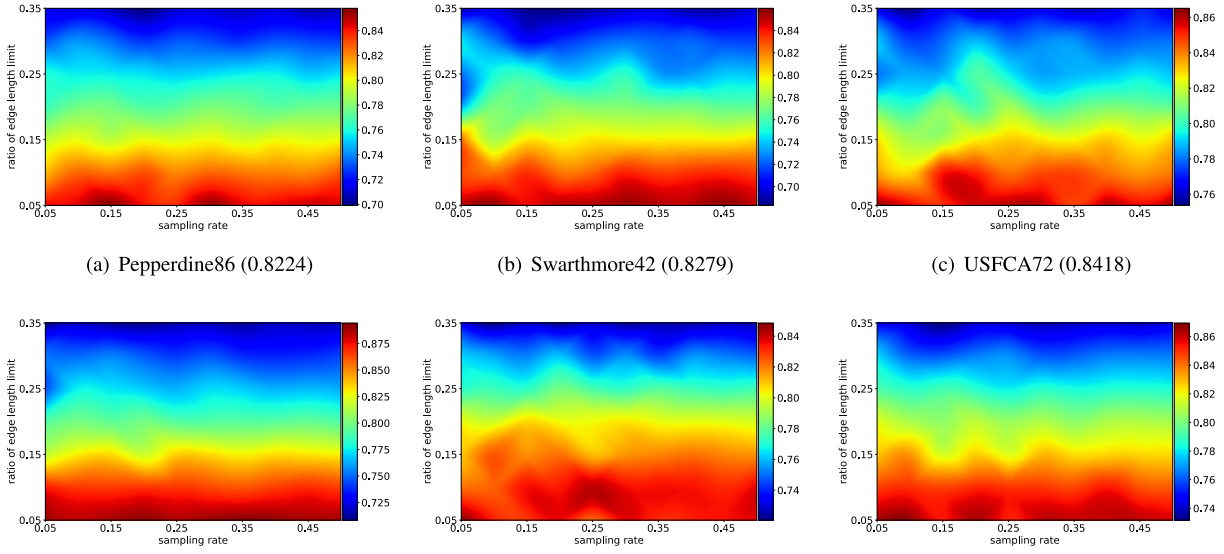
According to Fig.8(b), the value of diff_homophily does not change much when the edge length limit is the same. However, the length limit has significantly affected the homophily of the network. As shown in Fig.9, we selected some layout results with the same edge length limit and different sampling rates. Each case has a different edge number controlled by s. More topology information were revealed with a higher sampling rate.

Although homophily is not optimal under this parameter configuration, the higher the sampling rate, the better the layout results would be. Therefore, once the ratio of edge length limit that ensures relatively good homophily is set, users can determine the complexity of the network by controlling the sampling rate. In Fig.10, we listed other data's heatmap



(a) $s = 0, |E| = 961$ (b) $s = 0.05, |E| = 1176$ (c) $s = 0.25, |E| = 2043$ (d) $s = 0.45, |E| = 2909$

FIGURE 9. Layout results on Reed98($|V| = 962$, $|E| = 18812$) at different sample rates shows changes in network complexity.



(a) Pepperdine86 (0.8224) (b) Swarthmore42 (0.8279) (c) USFCA72 (0.8418)

(d) Wellesley22 (0.8618) (e) Mich67 (0.8282) (f) Williams40 (0.8449)

FIGURE 10. Diff_homophily heatmap of 6 different networks in Facebook100. Red area indicates a high value of diff_homophily, and the blue area is the opposite. The results on different networks are similar. A smaller edge length limit contributes more to diff_homophily, and the sampling rate does not have much effect on it. The average diff_homophily of each network is shown after the network name. It can be seen that almost selected edges are homophily edges.

of diff_homophily in the Facebook100 dataset. All the data heatmaps corresponded to the previous analysis that a smaller edge length limit contributes more to homophily, and the sampling rate does not have much effect on it.

4) EXPERIMENTS ON BSTRESS LAYOUT

In our method, the quality of the MST layout is essential. It initializes the vertex position and determines the edge selection to some extent. The bStress model allows the vertices to be evenly distributed across the plane, and such layout gives the IES-Backbone the desired results. Past methods only used the classical force-directed model for the layout.

We conclude that it is better to apply the bStress model in IES-Backbone rather than the force-directed model in the layout step for obvious reasons. Using the bStress model achieve improved visualization not only reflected in homophily metrics but also visual effects.

We run the bStress model and the force-directed model at the same time to get the MST layout, and the results were excessively different, which can be viewed in Fig.11(a) and Fig.11(b). The result of the bStress model barely shows edge crossing, and the tree structure is evenly spread out. It is a proper initialization for the next step. If there were too many edge crossings in the first step, the nodes from different clusters would place nodes from different cluster too close, which in turn would reduce homophily.

Fig.11(c) and Fig.11(d) show the comparison of the final layout results, and the bStress model has presented a more readable network. As shown in Fig.11(c), different clusters do not overlap much, and the proportion of heterophily edges is deficient. On the contrary, the result of the force-directed model shown in Fig.11(d) is confusing. Homophily metric also illustrates the advantage of the utilization of the bStress model. Fig.12 indicates that the homophily value of bStress model is generally higher

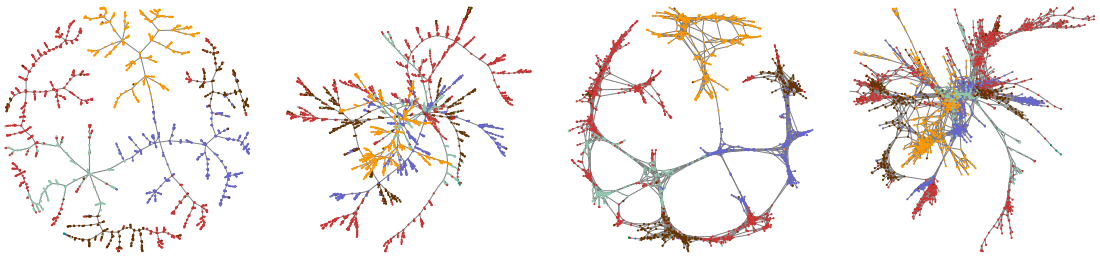
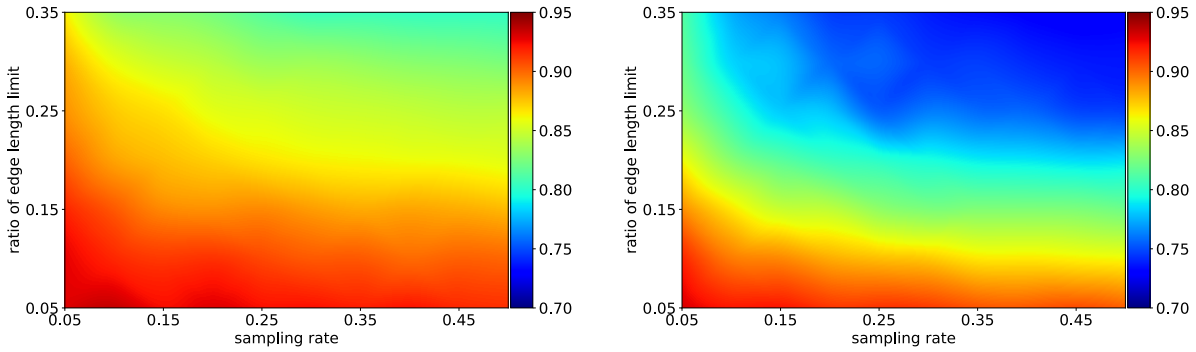


FIGURE 11. Results of bStress and force-directed model on Bowdoin47. (a) MST layout with the bStress model. (b) MST layout with force-directed model. The bStress model evenly distributes the MST get a better initialization. (c) Final layout result with the bStress model. (d) Final layout result with force-directed model. Force-directed model makes more visual clutter than bStress model.



(a) bStress model (b) force-directed model

FIGURE 12. The comparision of bStress model and force-directed model on homophily. Red area indicates a high value of homophily, and the blue area is the opposite. (a) Distributions of homophily in Bowdoin47 applied by bStress model. (b) Distributions of homophily in Bowdoin47 applied by force-directed model. Two heatmaps were colored in the same scale. As shown in the figures, the layout quality of bStress is much better than the force-directed model as the value in bStress model is much higher than the other.

than the force-directed model. The bStress model can be used to describe improved homophily metrics and visual result.

V. CONCLUSION

We proposed an IES-Backbone method to prevent the hairball layout when drawing a small world network. It solved two main problems in the contemporary approaches of loosing too much topology detail and low interactivity. We used an edge selection approach based on the distance of the graph to achieve a highly readable network layout. The complexity of the simplified network can be controlled by edge length limit and sampling rate so that users can interactively obtain results from the visualization system. We also used bStress model replacing the classic force-directed model in graph drawing to ensure the quality of edge selection step and the final result.

We evaluated the IES-Backbone method with visual results and metrics. As a result, the hairball layout was avoided, and the topology structure was displayed clearly. The average path length was less than other approaches, maintaining the topological detail pleasantly. The homophily of simplified network also performed ideally in the dataset of Facebook100. Most of the edges connected vertices from the same cluster. The two parameters are robust: the optimal value of edge length limit ratio is approximately the same in any network, and the value of the sampling rate is not sensitive to the homophily so that users can determine the complexity by controlling s without affecting the quality of the layout.

REFERENCES

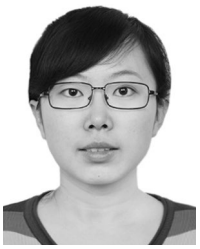
[1] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[2] M. Perc, J. J. Jordan, D. G. Rand, Z. Wang, S. Boccaletti, and A. Szolnoki, "Statistical physics of human cooperation," *Phys. Rep.*, vol. 687, pp. 1–51, May 2017.
[3] D. Helbing, D. Brockmann, T. Chadefaux, K. Donnay, U. Blanke, O. Woolley-Meza, M. Moussaid, A. Johansson, J. Krause, S. Schutte, and M. Perc, "Saving human lives: What complexity science and information systems can contribute," *J. Stat. Phys.*, vol. 158, no. 3, pp. 735–781, Feb. 2015.
[4] M. Perc, "The Matthew effect in empirical data," *J. Roy. Soc. Interface*, vol. 11, no. 98, Sep. 2014, Art. no. 20140378.
[5] C. McGrath, J. Blythe, and D. Krackhardt, "Seeing groups in graph layouts," *Connections*, vol. 19, no. 2, pp. 22–29, 1996.
[6] Q. H. Nguyen, S.-H. Hong, P. Eades, and A. Meidiana, "Proxy graph: Visual quality metrics of big graph sampling," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 6, pp. 1600–1611, Jun. 2017.
[7] B. Nick, C. Lee, P. Cunningham, and U. Brandes, "Simmelian backbones: Amplifying hidden homophily in Facebook networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2013, pp. 525–532.

[8] A. Nocaj, M. Ortmann, and U. Brandes, "Untangling the hairballs of multi-centered, small-world online social media networks," *J. Graph Algorithms Appl.*, vol. 19, no. 2, pp. 595–618, 2015.
[9] F. Van Ham and M. Wattenberg, "Centrality based visualization of small world graphs," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 975–982, 2008.
[10] A. Nocaj, M. Ortmann, and U. Brandes, "Adaptive disentanglement based on local clustering in small-world network visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 6, pp. 1662–1671, Jun. 2016.
[11] F. Zhou, S. Malher, and H. Toivonen, "Network simplification with minimal loss of connectivity," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 659–668.
[12] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1913–1926, 2011.
[13] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng, "Spectral sparsification of graphs: Theory and algorithms," *Commun. ACM*, vol. 56, no. 8, pp. 87–94, 2013.
[14] N. Ruan, R. Jin, and Y. Huang, "Distance preserving graph simplification," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 1200–1205.
[15] Y. Jia, J. Hoberock, M. Garland, and J. Hart, "On the visualization of social and other scale-free networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1285–1292, Nov. 2008.
[16] G. Simmel, *The Sociology of Georg Simmel*, vol. 92892. New York, NY, USA: Simon & Schuster, 1964.
[17] D. Edge, J. Larson, M. Mobius, and C. White, "Trimming the hairball: Edge cutting strategies for making dense graphs usable," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 3951–3958.
[18] H. Gibson, J. Faith, and P. Vickers, "A survey of two-dimensional graph layout techniques for information visualisation," *Inf. Vis.*, vol. 12, nos. 3–4, pp. 324–357, 2013.
[19] S.-H. Cheong and Y.-W. Si, "Force-directed algorithms for schematic drawings and placement: A survey," *Inf. Vis.*, Jan. 2019, doi: 10.1177/1473871618821740.
[20] P. Eades, "A heuristic for graph drawing," *Congressus Numerantium*, vol. 42, pp. 149–160, 1984.
[21] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, 1989.
[22] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw., Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.
[23] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "ForceAtlas2, A continuous graph layout algorithm for handy network visualization designed for the Gephi software," *PloS One*, vol. 9, no. 6, 2014, Art. no. e98679.
[24] J. De Leeuw and G. Michailidis, "Graph layout techniques and multi-dimensional data analysis," in *Game Theory, Optimal Stopping, Probability and Statistics* (Lecture Notes-Monograph Series). JSTOR, 2000, pp. 219–248.
[25] E. R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," in *Proc. Int. Symp. Graph Drawing*. Berlin, Germany: Springer, 2004, pp. 239–250.
[26] Y. Koren and A. Çivril, "The binary stress model for graph drawing," in *Proc. Int. Symp. Graph Drawing*. Berlin, Germany: Springer, 2008, pp. 193–205.
[27] M. Behrisch, M. Blumenschein, N. W. Kim, L. Shao, M. El-Assady, J. Fuchs, D. Seebacher, A. Diehl, U. Brandes, H. Pfister, T. Schreck, D. Weiskopf, and D. A. Keim, "Quality metrics for information visualization," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 625–662, 2018.
[28] Q. Nguyen, P. Eades, and S.-H. Hong, "On the faithfulness of graph visualizations," in *Proc. Int. Symp. Graph Drawing*. Berlin, Germany: Springer, 2012, pp. 566–568.
[29] C. Dunne, S. I. Ross, B. Shneiderman, and M. Martino, "Readability metric feedback for aiding node-link visualization designers," *IBM J. Res. Develop.*, vol. 59, nos. 2–3, pp. 14:1–14:16, Mar./May 2015.
[30] M. D. Humphries and K. Gurney, "Network 'small-world-ness': A quantitative method for determining canonical network equivalence," *PLoS One*, vol. 3, no. 4, 2008, Art. no. e0002051.
[31] Z. P. Neal, "How small is it? Comparing indices of small worldliness," *Netw. Sci.*, vol. 5, no. 1, pp. 30–44, 2017.
[32] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
[33] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.
[34] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos Nat. Lab., Los Alamos, NM, USA, Tech. Rep. LA-UR-08-5495, 2008.
[35] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter, "Comparing community structure to characteristics in online collegiate social networks," *SIAM Rev.*, vol. 53, no. 3, pp. 526–543, 2011.
[36] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, pp. 155–168, 2008.
[37] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, 1983.

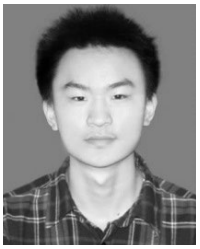
CHENG ZHAN received the bachelor's degree from the Harbin Institute of Technology (HIT) at Weihai, China, in 2017. He is currently pursuing the master's degree with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. His research interests include network visualization and visual analysis.

DAOBING ZHANG received the B.Sc. degree in communication and signal processing and the Ph.D. degree in optics engineering from the Graduate University of the Chinese Academy Sciences, Beijing, China, in 2004 and 2007, respectively.

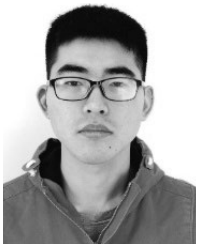
He is currently a Researcher with the Key Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing. His research interests include high resolution remote sensing image processing and interpretation.



YANG WANG received the B.E. degree from Beihang University, Beijing, China, and the Ph.D. degree from Peking University, Beijing, China. She is currently an Assistant Professor with the Aerospace Information Research Institute, Chinese Academy of Sciences. Her research interests include the geospatial data organization and visualization.



DAOYU LIN received the bachelor's degree from the Beijing University of Posts and Telecommunications. He is currently pursuing the master's degree with the Aerospace Information Research Institute, Chinese Academy of Sciences, working on computer vision and image processing.



HUI WANG received the bachelor's degree from the Hebei University of Technology at Tianjin, China, in 2018. He is currently pursuing the master's degree with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. His research interests include image processing and deep learning.