Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2015, Article ID 105385, 11 pages
http://dx.doi.org/10.1155/2015/105385

Research Article

Extracting Backbones from Weighted Complex Networks with Incomplete Information

Liqiang Qian,[1] Zhan Bu,[2] Mei Lu,[1] Jie Cao,[2] and Zhiang Wu[2]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *Jiangsu Provincial Key Lab. of E-Business, Nanjing University of Finance and Economics, Nanjing 210046, China*

Correspondence should be addressed to Zhan Bu; buzhan@nuaa.edu.cn

Received 16 July 2014; Revised 21 September 2014; Accepted 21 September 2014

Academic Editor: Zidong Wang

The backbone is the natural abstraction of a complex network, which can help people understand a networked system in a more simplified form. Traditional backbone extraction methods tend to include many outliers into the backbone. What is more, they often suffer from the computational inefficiency—the exhaustive search of all nodes or edges is often prohibitively expensive. In this paper, we propose a backbone extraction heuristic with incomplete information (BEHwII) to find the backbone in a complex weighted network. First, a strict filtering rule is carefully designed to determine edges to be preserved or discarded. Second, we present a local search model to examine part of edges in an iterative way, which only relies on the local/incomplete knowledge rather than the global view of the network. Experimental results on four real-life networks demonstrate the advantage of BEHwII over the classic disparity filter method by either effectiveness or efficiency validity.

1. Introduction

Complex networks have become an important approach for understanding systems involving interacting objects [1]. Thus, networked systems have permeated a wide spectrum of domains, ranging from the biology and the automatic control to the computer science [2, 3]. With networked systems being increasingly large, to understand and reveal the underlying phenomena taking place in such systems are facing considerable challenges. The presence of the backbone is a signature or an abstraction of the nature of complex systems and can provide huge help for understanding them in more simplified forms [4]. For example, detecting the backbones in criminal networks can better target suspects [5]. Also, urban planners attempt to examine the topologies of public transport systems by analyzing their backbones [6].

Recent years have witnessed an increasing interest in extracting backbones in large-scale weighted networks of various kinds [4, 7–9]. As many networks are evolving into large scale and the weight distributions are spanning several orders of magnitude, extracting backbones from them has become a critical task for research and applications of various purposes. In general, the backbone should be thought of as a set of nodes and edges that interconnect various pieces of network, providing a path for the exchange of information between different subnetworks [10]. Thus, a promising way for backbone extraction is to map the original network into a smaller network, in which the numbers of nodes and edges should be small enough to be amenable to analysis and visualization.

In the literature, the existing methods can be roughly divided into two categories, one based on the coarse graining and the other is filter-based. The methods based on the coarse graining [4, 7, 11–14] clump nodes sharing common attributes together in the same group/community and then consider the whole group as one single unit in the new networks. However, there is often no clear statement on whether properties of the initial network should be preserved in the network of clusters [15].

The filter-based methods [8, 9, 16–18] typically employ a bottom-up strategy to extract the backbone. They often start by defining a statistical property of a node or an edge, and then this property is used as a criterion to determine nodes/edges to be preserved or discarded. In this case,

the observation scale is fixed and the representation that the network symbolizes is not changed. Instead, those elements, nodes, and edges, which carry relevant information about the network structure, are kept while the rest are discarded. However, the filter-based methods may include a multitude of outliers, which should not be included into the backbone naturally. What is more, they often suffer from the computational inefficiency: the exhaustive search of all nodes or edges is often prohibitively expensive.

In this work, we attempt to design a novel filter-based method for extracting backbones from large-scale weighted networks. Unlike the exhaustive search adopted by the existing methods, the proposed approach only needs incomplete information and then invokes the iteratively local search scheme for improving the efficiency. So, this novel method is called backbone extraction heuristic with incomplete information (BEHwII). In particular, although α_{ij} proposed in [8] is employed as the filtering criterion, BEHwII imposes max instead of min to enhance the filtering rule, so that the case of extracting too many outliers into the backbone can be avoided. Our method is naturally a heuristic, since it does not examine all edges in the network. Alternatively, BEHwII greedily selects an optimal edge in one iteration and adds this edge into the backbone if the predefined max filtering rule is satisfied. Extensive experiments on various real-world networks demonstrate the superiority of BEHwII over the global filtering method in terms of effectiveness and efficiency.

The remainder of this paper is organized as follows. In Section 2, we introduce preliminaries and motivation of this work. In Section 3, we discuss the local search mechanism and then present the algorithmic details of BEHwII. Experimental results will be given in Section 4. We present the related work in Section 5 and finally conclude this paper in Section 6.

2. Preliminaries and Motivation

Since the proposed method for backbone extraction is a filter-based model in essence, we begin by providing the preliminary knowledge about the filter-based model. Thus, we analyze some drawbacks of existing filter-based methods, which leads to a better understanding of the motivation of this paper.

The filter-based models typically employ a bottom-up strategy to extract the backbone. They often start by defining a statistical property of a node or an edge, and then this property is used as a criterion to determine nodes/edges to be preserved or discarded. As a result, preserved nodes and their links, or preserved edges and their endpoints, composed the backbone of the network. Therefore, the key step in filter-based methods is how to define a reasonable filtering property for nodes/edges. For instance, k-core is a well-known filtering property that is used to construct a hierarchical topological filter in [16]. However, many simple filtering properties (e.g., k-core) are not suitable for weighted networks. Meanwhile, the real-world weighted networks are usually with strong disorder heavy-tailed distributions of weights [19]. That is, the probability distribution $P(w)$ that any given link carries a weight w is broadly distributed, spanning several orders of magnitude. This feature exerts nontrivial challenges to define the filtering property for weighted networks, due in large part to the lack of a characteristic scale. Serrano et al. [8] addressed this challenge by introducing the disparity filter based on the null hypothesis; that is, the normalized weights that correspond to the connections of a certain node of degree k are produced by a random assignment from a uniform distribution. Given a node i and its associated link with weight w_{ij}, the normalized weight p_{ij} is defined as

$$p_{ij} = \frac{w_{ij}}{\sum_l w_{il}}. \tag{1}$$

Under the null hypothesis, a *null model* is then presented, in which $k-1$ points are distributed with uniform probability in the interval $[0, 1]$. As a result, k subintervals are generated, of which lengths represent the expected values for the k normalized weights p_{ij} according to the null hypothesis. The probability density function for one of these variables taking a particular value x is

$$\rho(x)\,dx = (k-1)(1-x)^{k-2}\,dx. \tag{2}$$

Based on (2), given an edge, the probability α_{ij} indicating its normalized weight p_{ij} is compatible with the null model and can be defined as

$$\begin{aligned}\alpha_{ij} &= 1 - (k_i - 1)\int_0^{p_{ij}} (1-x)^{k_i-2}\,dx \\ &= \left(1 - p_{ij}\right)^{k_i-1},\end{aligned} \tag{3}$$

where k_i is the degree of node i. Thus, α_{ij} is adopted as the filtering criterion in [8] for weighted networks. Given a significance level α, the edges that carry weights which can be considered not compatible with a random distribution can be filtered out with a certain statistical significance. That is, edges with $\alpha_{ij} < \alpha$ should be kept, since they reject the null hypothesis.

The criterion α_{ij} gave birth to an effective filter-based method for backbone extraction [8]. However, two drawbacks have attracted our attention. One of the biggest limitations is that it may include a multitude of outliers, which should not be included into the backbone naturally. In what follows, we try to explore its cause and give a modified scheme.

For node i with degree k_i, the level of local heterogeneity in the weights can be calculated as

$$\gamma(k_i) = k_i \sum_j p_{ij}^2. \tag{4}$$

Thus, under perfect homogeneity, when all the links share the same amount of the strength of the node, $\gamma(k_i)$ equals 1 independently of k_i, while in the case of perfect heterogeneity, when just one of the links carries the whole strength of the node, $\gamma(k_i)$ is equal to k_i. With predefined null model, the join probability distribution for two intervals can be defined as

$$\begin{aligned}\rho(x, y)\,dx\,dy = (k-1)(k-2)\left(1-x-y\right)^{k-3}\Theta \\ \times (1-x-y)\,dx\,dy,\end{aligned} \tag{5}$$

where $\Theta(\cdot)$ is the Heaviside step function, which can be used to calculate the statistics of $\gamma_{\text{null}}(k_i)$ for the null model. The average $\mu(\gamma_{\text{null}}(k_i))$ and the standard deviation $\sigma^2(\gamma_{\text{null}}(k_i))$ are estimated to be

$$\mu\left(\gamma_{\text{null}}\left(k_i\right)\right) = \frac{2k_i}{k_i+1},$$
$$\sigma^2\left(\gamma_{\text{null}}\left(k_i\right)\right) = k_i^2\left(\frac{4k_i+20}{\left(k_i+1\right)\left(k_i+2\right)\left(k_i+3\right)} - \frac{4}{\left(k_i+1\right)^2}\right). \tag{6}$$

In real networks, the observed level of local heterogeneity, denoted by $\gamma_{\text{ob}}(k_i)$, can be compared against the null model expectations. Namely, the observed values are compatible with the null hypotheses when they lie between the perfect homogeneity and $\mu(\gamma_{\text{null}}(k_i)) + a \cdot \sigma(\gamma_{\text{null}}(k_i))$. And the local heterogeneity will be recognized only if $\gamma_{\text{ob}}(k_i)$ obeys

$$\gamma_{\text{ob}}\left(k_i\right) > \mu\left(\gamma_{\text{null}}\left(k_i\right)\right) + a \cdot \sigma\left(\gamma_{\text{null}}\left(k_i\right)\right). \tag{7}$$

The parameter a is a constant determining the confidence interval for the evaluation of the null hypothesis. The larger it is, the more restrictive the null model becomes and the more disordered weights should be for local heterogeneity to be detected. A typical value of a in analogy to Gaussian statistics could be set as 2. In Figure 1, we show two regions (local heterogeneity and local compatibility) associated with different k_i. Obviously, small nodes in terms of degree (e.g., $k_i < 5$) are more likely to fall into the local compatible region, which implies that those nodes with small degree should not be preserved in the backbone.

In [8], the multiscale backbone is obtained by preserving all the links which beat the significant level α for at least one of the two nodes at the ends of the link while discounting the rest. Notice that α_{ij} is not symmetrical; that is, $\alpha_{ij} \neq \alpha_{ji}$, if $k_i \neq k_j$. In the case of a node i with degree $k_i < 5$ connected to a node j with degree $k_j \gg 5$, we might have $\alpha_{ji} < \alpha < \alpha_{ij}$. Then this link will be preserved as it holds $\min(\alpha_{ij}, \alpha_{ji}) < \alpha$. However, as discussed above, node i is likely to fall into the local compatible region, which should be kept away from the backbone. Considering that an intermediate power law degree distribution is usually observed in real systems, the disparity filter in [8] may include a multitude of outliers. To avoid including many outliers into the backbone, one can impose max instead of min to enhance the filtering rule, so that a connection is preserved whenever its intensity is significant for both nodes involved.

Secondly, most of the existing filter-based methods [8, 9, 16, 17] suffer from the computational inefficiency, the exhaustive search of all nodes or edges in a network. For example, the filtering method based on α_{ij} is heavily dependent on the number of links. As many social networking sites are evolving into superlarge scales, for example, containing millions even billions of nodes and edges, the computation will be terrible!

According to the above analysis, this paper proposes a local method for extracting backbones from weighted networks. In particular, we try to answer the following two questions:

(i) Q1: how to carefully design a filtering criterion to avoid including many outliers into the backbone?

(ii) Q2: how to reduce the computational complexity of the backbone extraction algorithm?

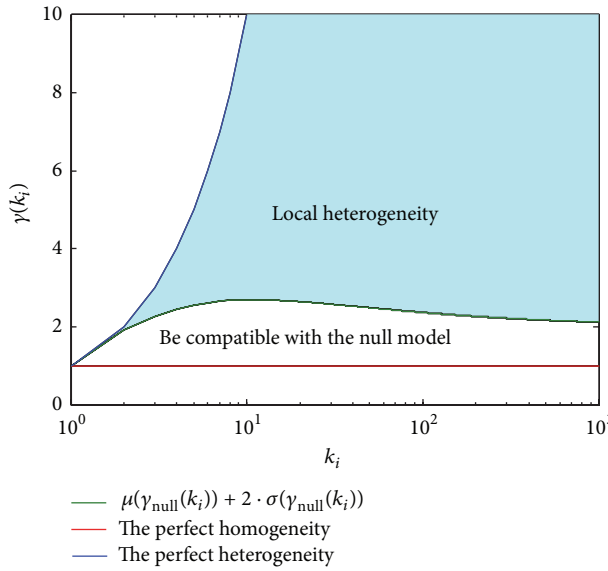


FIGURE 1: $\gamma(k_i)$ compared against the null model expectations.

3. Backbone Extraction Heuristic with Incomplete Information (BEHwII)

Let $\mathcal{G} = (V, E, W)$ be a given weighted graph, where V is the set of nodes ($|V| = n$), E is the set of edges ($|E| = m$) that connect the nodes in V, and W is the weight of every edge in E. Backbone extraction is formulated as finding a subset of graph $\mathcal{G}' = (V', E')$, that is, the backbone, where $|E'| \ll |E|$ and $\forall e_{ij} \in E'$, $\alpha_{ij} < \alpha$. This implies that the backbone should also significantly reduce the number of edges, while preserving most essential connections.

In this section, we propose a backbone extraction heuristic with incomplete information (BEHwII for short). First, we introduce the basic idea of BEHwII, covering the local search mechanism. Second, we present algorithmic details including the complexity analysis for BEHwII.

3.1. Local Search Model. In this paper, we employ the filtering criterion α_{ij} proposed in [8]. However, one major drawback lies in that it is probable to include too many outliers into the backbone as stated in Section 2. To explore its cause, we argue that this drawback originates from the looseness of the filtering rule, that is, $\min(\alpha_{ij}, \alpha_{ji}) < \alpha$. Therefore, BEHwII attempts to impose max instead of min to enhance the filtering rule, so that a connection is preserved whenever its intensity is significant for both nodes involved. In BEHwII, an edge e_{ij} is *preserved* in the backbone, if

$$\alpha_{ij}^* = \max\left(\alpha_{ij}, \alpha_{ji}\right) < \alpha, \tag{8}$$

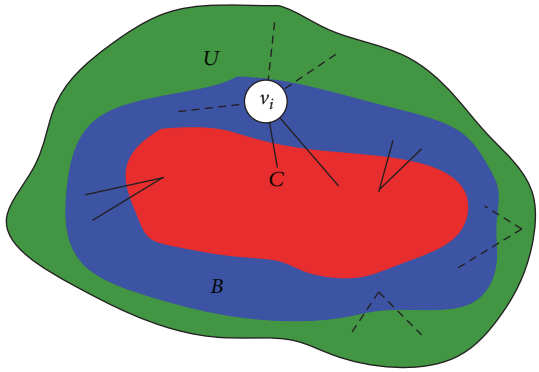


Figure 2: Illustration for the local search.

where α_{ij} is the probability derived by comparing the normalized weight p_{ij} with the null model, as shown in (3). With the filtering rule, BEHwII aims to extract a certain percentage (denoted by $\%m_t$) of edges satisfying (8) as the backbone.

A straightforward way for backbone extraction is to apply the exhaustive search, that is, to examine all of the edges one by one, and add the edge to the backbone as (8) satisfied. Obviously, this exhaustive search suffers from the computational inefficiency, especially when the network becomes much larger. Here, we introduce a local search model to solve this problem. We divide the explored graph into three regions: the known local area $\mathscr{C}$, the boundary area $\mathscr{B}$, and a larger unknown area $\mathscr{U}$, as illustrated in Figure 2. Initially, we randomly select a node v_s as the start node and add v_s to $\mathscr{C}$. Then, all neighbors of nodes in $\mathscr{C}$ (e.g., v_s) are added to $\mathscr{B}$. The local search model selects an optimal edge e_{ij} with minimum α^*_{ij} from $\mathscr{C} \cup \mathscr{B}$ and adds it into the backbone if it holds (8). Areas $\mathscr{C}$ and $\mathscr{B}$ are expanded accordingly. Another edge will be selected and checked, until a certain number of edges are included into the backbone.

Remark 1. The local search model is a streaming and iterative scheme in essence [20]. An iterative process is invoked to examine each node along with its neighbors and performs a computation, of which the result is associated with the processed node. Such scheme is a very promising technique of scaling the existing method. Moreover, the local search model is independent of the "global knowledge"; that is, it only needs to fetch part of the node adjacency lists into main-memory. Due to the small-world effect, our model is validated to be slightly dependent on the initial node selection, of which the experimental results will be given in Section 4.1.

3.2. Algorithmic Details. In this section, we introduce how to use BEHwII to extract the backbone starting from any randomly selected node. BEHwII initially places the randomly selected source node v_s into the known local area ($\mathscr{C} \leftarrow \{v_s\}$) and adds its neighbors into $\mathscr{B}$. Two data structures used in BEHwII are described as follows:

(i) *Min-heap* H, which stores the edge information, including e_{ij} and $\max(\alpha_{ij}, \alpha_{ji})$, in $\mathscr{C} \cup \mathscr{B}$, so that every update process will take $O(\log |H|)$ time;

```
(1)  procedure BEHwII(v_s, α, %m_t)
(2)      C ← {v_s};
(3)      B ← {v_i | v_i ∈ N_s};
(4)      H ← {⟨e_si, α*_si⟩}, where α*_si = max(α_si, α_is);
(5)      while |L^E| ≤ %m_t * m do
(6)          Get the minimal α*_ij from H;
(7)          if α*_ij < α then
(8)              L^E ← e_ij;
(9)          end if
(10)         H ← H/e_ij;
(11)         if ∃i' ∈ {i, j}, v_i' ∉ C then
(12)             C ← C ⋃ {v_i'};
(13)             B ← B ⋃ {v_j' | v_j' ∈ N_i'};
(14)             H ← H ⋃ {⟨e_i'j', α*_i'j'⟩};
(15)         end if
(16)         if |C| ≥ n then
(17)             break;
(18)         end if
(19)     end while
(20)     return L^E;
(21) end procedure
```

Algorithm 1: BEHwII algorithm.

(ii) *List* L^E, which stores the edges of the backbone, and every insert process will take $O(1)$ time.

We describe the BEHwII Algorithm step by step roughly as follows.

Step 1. Find the edge e_{ij} with the minimal value of $\max(\alpha_{ij}, \alpha_{ji})$ in $\mathscr{C} \cup \mathscr{B}$ and add it into L^E if it satisfies (8).

Step 2. If any endpoints on the considered edge e_{ij} are not included in $\mathscr{C}$ ($\exists i' \in \{i, j\}, v_{i'} \notin \mathscr{C}$), remove $v_{i'}$ from $\mathscr{B}$ to $\mathscr{C}$; otherwise, delete edge e_{ij} and turn to Step 1.

Step 3. Delete edge e_{ij} and remove additional nodes ($v_{j'} \mid v_{j'} \in N_{i'}, v_{j'} \in \mathscr{U}$) from $\mathscr{U}$ to $\mathscr{B}$.

The above process continues until it has agglomerated a certain percentage of edges, or it has discovered the entire enclosing component, whichever happens first. Note that if e_{ij} with the minimal value of $\max(\alpha_{ij}, \alpha_{ji})$ in Step 1 does not satisfy (8), we still check its endpoints and add corresponding edges into $\mathscr{C} \cup \mathscr{B}$. Here, the nodes between e_{ij} can be seen as the excessive nodes to continue the search process. See Algorithm 1 for more exact pseudocode.

Computational Complexity. The main computational cost of the above algorithm originates from the number of examined edges M. For each examined edge e_{ij}, BEHwII needs to calculate the value of $\max(\alpha_{ij}, \alpha_{ji})$ on it and update the min-heap H. Because $\max(\alpha_{ij}, \alpha_{ji})$ depends on the degrees of nodes v_i and v_j and on the normalized weights p_{ij} and p_{ji}, thus, it takes $O(k_i + k_j)$ time to calculate $\max(\alpha_{ij}, \alpha_{ji})$ on each examined edge. The updating (inserting or deleting) cost of H for each examined edge is $O(\log |H|)$. In general, the running

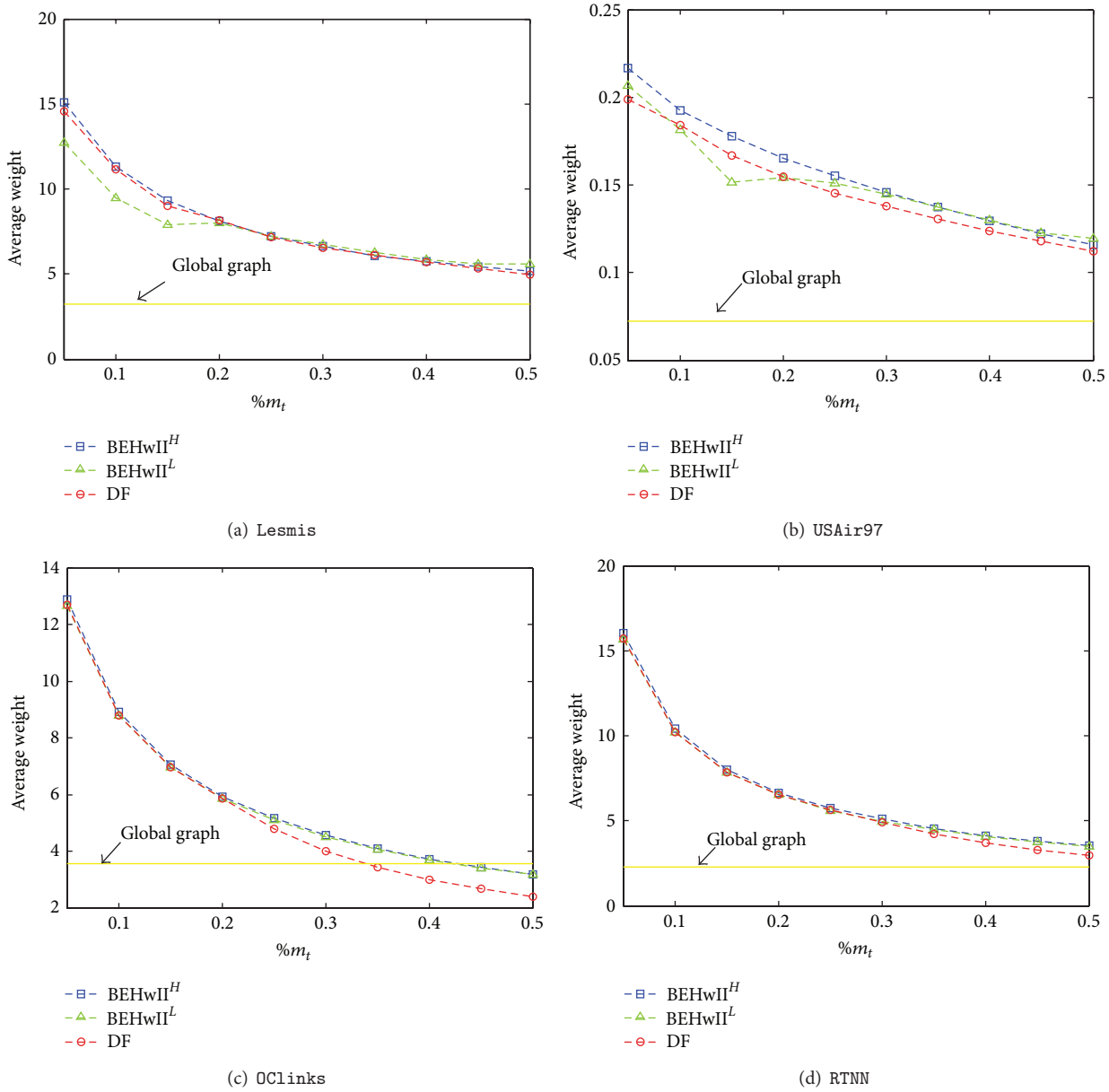


Figure 3: Comparison in terms of the average weight.

Table 1: Real-world networks for experiments.

Network	$\|V\|$	$\|E\|$	$\langle k\rangle$	$\langle w\rangle$
Lesmis	77	254	6.60	3.22
USAir97	322	2,126	12.80	0.07
OClinks	1,899	20,296	14.60	2.95
RTNN	13,308	148,035	22.25	2.29

time for the algorithm is $O(M(2\langle k\rangle + \log|H|))$, where $\langle k\rangle$ is the average degree of the graph.

4. Experimental Results

Four real-world undirected and weighted networks, `Lesmis`, `USAir97`, `OClinks`, and `RTNN`, are used for experiments. Some characteristics of these networks are shown in Table 1, where $|V|$ and $|E|$ indicate the numbers of nodes and edges, respectively, in the network, $\langle k\rangle$ indicates the average degree, and $\langle w\rangle$ indicates the average weight. `Lesmis` [21] is the network of coappearances of characters in Victor Hugo's novel, where nodes represent characters and edges connect any pair of characters that appear in the same chapter of the book. `USAir97` [22] gathers 2126 flight information between 332 US airports, where the weight represents the normalized distance among two airports. `OClinks` [23] is a network created from an online community, where nodes represent students at the University of California and edges are established between two students if one or more messages have been sent from one to the other. `RTNN` [24] is also a coappearance network including all words/terms in online

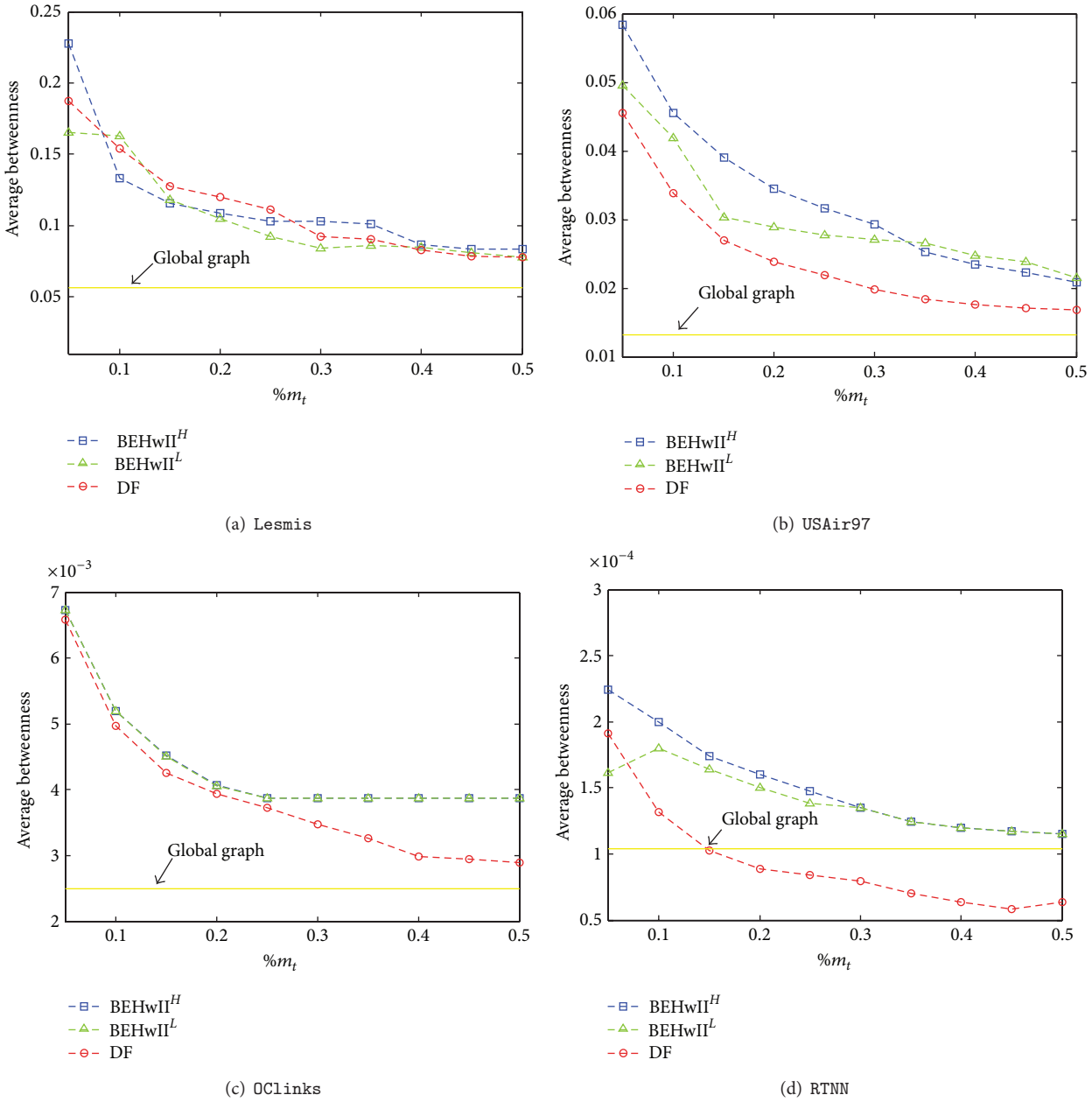
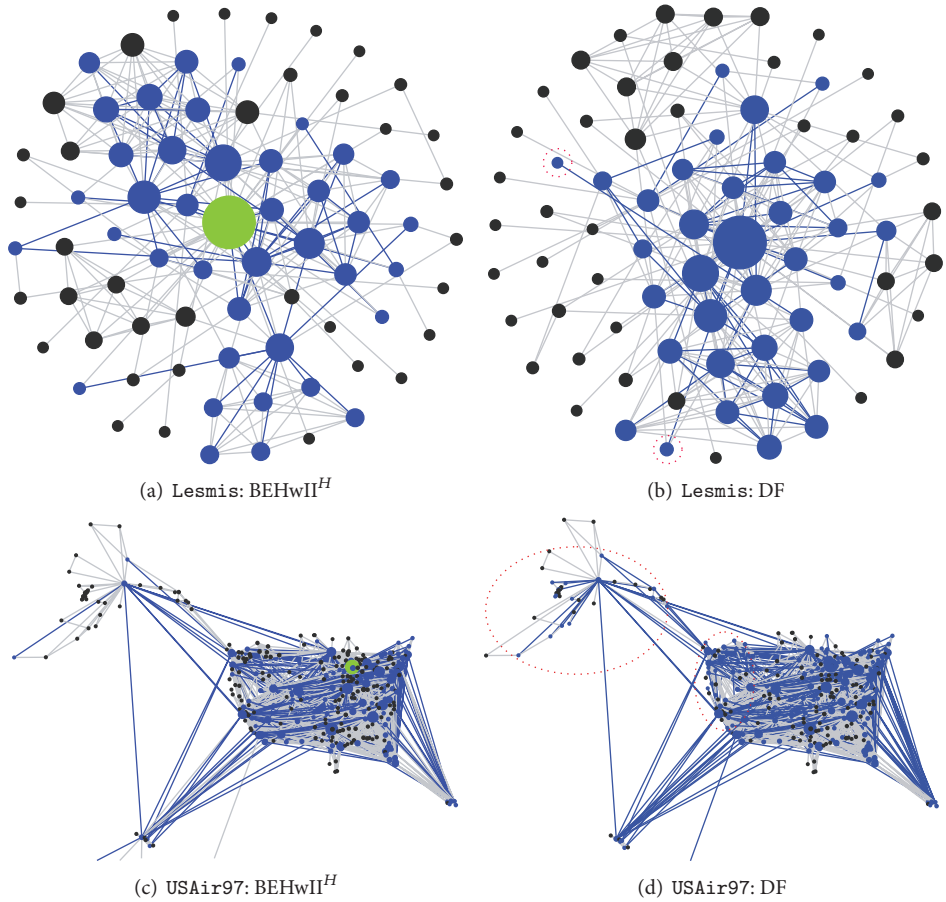


Figure 4: Comparison in terms of the average betweenness.

stories about the September 11 attack, where each node represents a word and each tie means that the two words appear in the same story.

4.1. Comparison Results. In this subsection, we compare BEHwII with the disparity filter (DF for short) proposed by Serrano et al. [8] in performance and scalability. BEHwII is a local search based algorithm, which can start from any randomly selected source node. To investigate the impact of the parameter v_s, we fix $\alpha = 0.5$ and take $v_s = v_h, v_l$, respectively, where v_h is a high-connected node and v_l is a low-connected one. Both v_h and v_l are randomly selected from the original network. For convenience, we denote BEHwII starting from v_h by BEHwII^H; then BEHwII^L represents BEHwII starting from v_l. For a given extraction goal (the percent edges kept in the backbone), the effectiveness of BEHwII^H, BEHwII^L, and DF can be validated by measuring the average weight and node betweenness of the extracted backbones, while the efficiencies can be measured by the number of examined edges and the overall running time.

Effectiveness. Figure 3 shows the average weight of the extracted backbones when the original graphs are extracted by BEHwII^H, BEHwII^L, and DF, respectively. Note that as the only parameter for DF is α, for a given network, the fraction of extracted edges $\%m_t$ is a monotonically increasing function



(a) Lesmis: BEHwII^H (b) Lesmis: DF

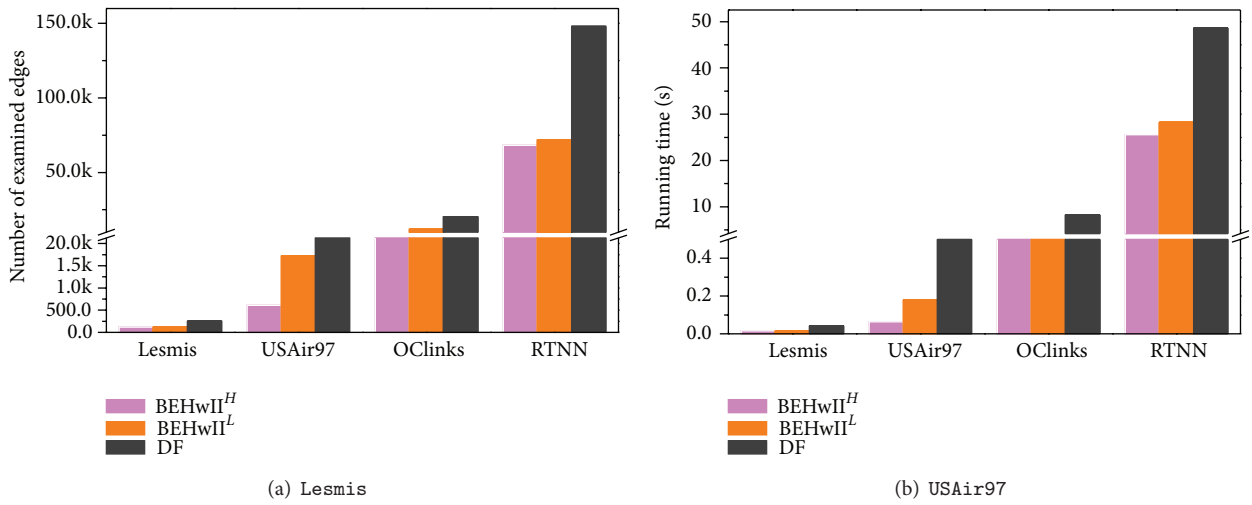
(c) USAir97: BEHwII^H (d) USAir97: DF

Figure 5: Comparison in terms of network visualizations.

of α. For convenient comparison, both DF and BEHwII use the same parameter $\%m_t$, which is gradually increased so that the number of extracted edges grows accordingly. Two observations are noteworthy from Figure 3. First, compared with DF, BEHwII^H shows slight improvements in terms of the average weight, no matter what $\%m_t$ is input. BEHwII^L does not perform well when $\%m_t$ is set to be too small. For instance, BEHwII^L obtains the $\%m_t = 0.1$ backbone with the average weight lower than 10 on the Lesmis network, but, after using BEHwII^H and DF to extract backbones, the average weight increases significantly. Another important observation is that BEHwII^H and BEHwII^L will trend consistently as $\%m_t$ grows to a certain level. As can be seen from Figures 3(a) and 3(b), when the fraction of edges grows to around 0.25, the backbones extracted by BEHwII^H and BEHwII^L will have the same value of average weight. As BEHwII^L adds local optimum edge into the backbone, even if it starts from a low-connected source node, it can sniff several high-connected nodes within limited steps. Therefore, BEHwII^L will evolve to a BEHwII^H after a certain percentage of edges have been discovered.

We then extensively explore the average node betweenness in the backbones extracted from Lesmis, USAir97, OClinks, and RTNN. Node betweenness centrality is the fraction of all shortest paths in the network that contain a given node, which reflects the connectedness of the node. Figure 4 shows the average betweenness of extracted nodes for different fractions of edges $\%m_t$ in the backbones. We can clearly find out that both BEHwII^H and BEHwII^H outperform DF in all of the test graphs. This implies that the edges extracted by BEHwII always lie between two high-connected nodes. As for DF, the filtering rule is so loose that some outliers (nodes with degree equal 1) will be included in the backbones, which will drop the connectedness of extracted backbone.

We then take a direct look at the extracted backbones. The Lesmis and USAir97 networks are used here as two examples. We set $\%m_t = 0.25$ and $\alpha = 0.5$ for BEHwII^H. In the case of Lesmis, the extracted backbone obtained by BEHwII^L is shown in Figure 5(a). The source node is colored with green, the nodes and edges colored with blue are those kept in the backbones, the size of the node expresses its strength ($\sum_l w_{il}$), and the thickness of the edge represents the weight on it. Interestingly, the backbone obtained by BEHwII^H preserves almost all high-connectivity nodes and essential connections. We then employ DF directly on this network and obtain a backbone as shown in Figure 5(b). The clique-like pattern on the top is missed, and, what is more, two outliers (highlighted by dashed circles) are kept.



(a) Lesmis (b) USAir97

Figure 6: Comparison on efficiency.

As for the USAir97 network, nodes are placed in the plane according to their actual coordinates on the earth. The backbone extracted by BEHwII^H, as shown in Figure 5(c), almost covers all the geographic regions of USA. In addition, the hierarchy of the transportation system is fully highlighted, including not just the most high flux connections but also small weight edges that are statistically significant because they represent relevant signal at the small scales. However, the backbone extracted by DF includes many small airports in Alaska and the west coast of USA (highlighted in dashed ellipses).

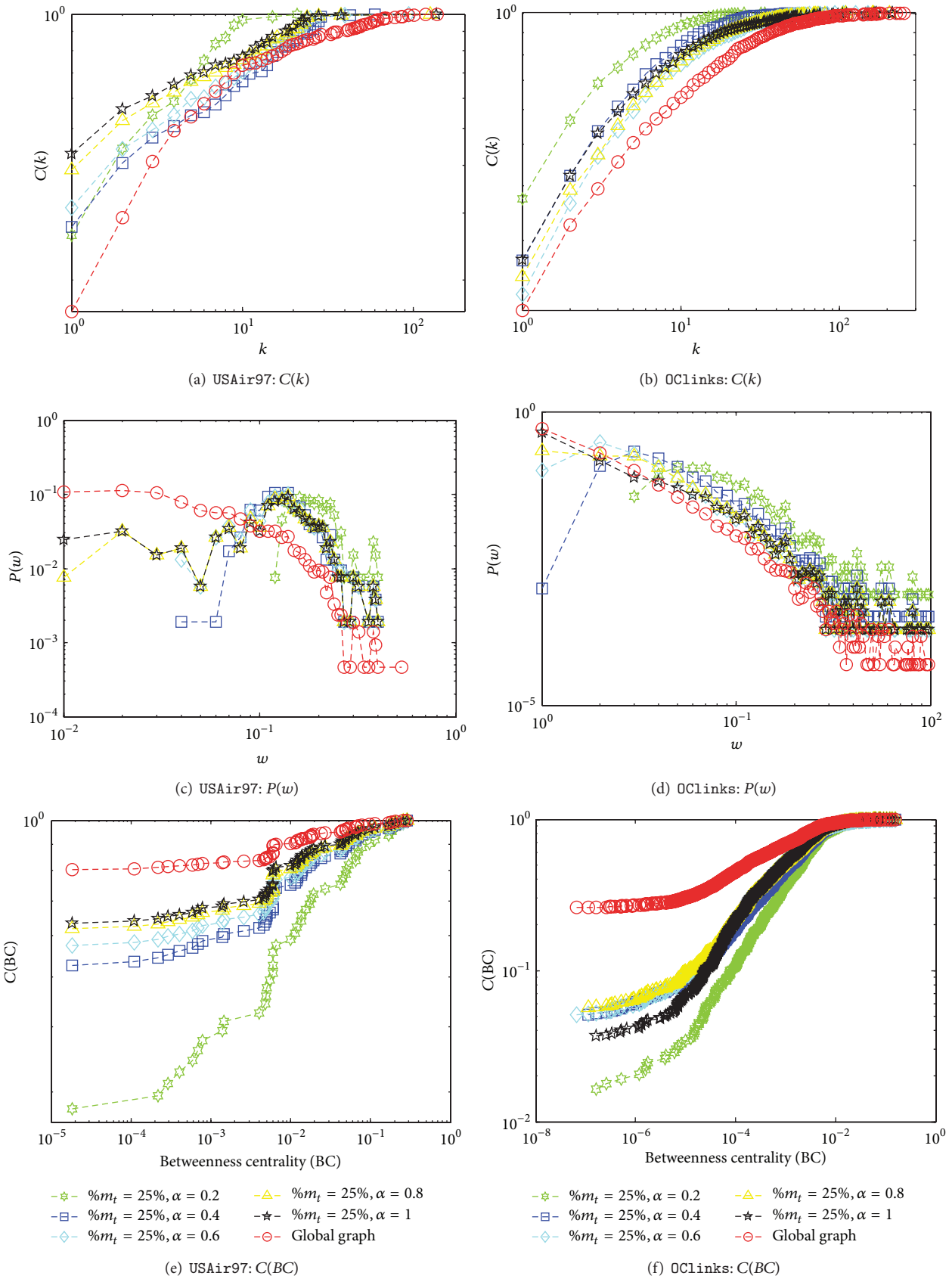
The Efficiency. Figure 6 compares the efficiencies of BEHwII and DF, given the extraction goal $\%m_t = 0.25$. The numbers of examined edges by BEHwII^H, BEHwII^L, and DF for the four test networks are shown in Figure 6(a). Apparently, BEHwII^H and BEHwII^L examine fewer edges than DF does. The latter will examine all nodes and edges in the network. Figure 6(b) verifies our analysis in Section 3.2; that is, the running time of BEHwII originates from the number of examined edges. It is interesting to find that the running time of BEHwII^H and BEHwII^L remains nearly constant in relative large dense graphs (e.g., OClinks and RTNN), that is, because those two networks have the "small world" effect [23, 24], in which most nodes can be reached from each other by a small number of hops or steps. In this context, both BEHwII^H and BEHwII^L can rapidly sniff those high-connected nodes; therefore their overall running times are almost consistent.

4.2. Inside BEHwII. Here, we take a further step to explore several factors that affect the performance of BEHwII. We select BEHwII starting from a high-connected source node, that is, BEHwII^H, for experiments. Two inside factors have been investigated: the significant level α and the inside filtering rule.

The Significant Level α. It is particularly interesting to analyze the behavior of the topological properties of the backbones extracted by BEHwII^H at increasing levels of the significant level α. Figures 7(a) and 7(b) show the evolution of the cumulative degree distribution, $C(k) = \sum_{k' \le k} P(k')$, with different values of α for USAir97 and OClinks, respectively. The backbones extracted by BEHwII^H have the cumulative degree distributions similar to the original networks. Smaller values of α have flat startups, indicating that the extracted backbones contain fewer low-degree nodes. The evolution of the weight distribution ($P(w)$) with different values of α is shown in Figures 7(c) and 7(d), from which we observe that the original USAir97 and OClinks networks are both heavy tailed. Interestingly, almost all scales are kept during the search process until BEHwII^H becomes too restrictive, in which case BEHwII^H applies a very small value of α. A restrictive BEHwII^H cuts $P(w)$ off below w_c, which may discard the region of small weights. Finally, we analyze the cumulative node betweenness centrality distributions of extracted backbones. It is worth mentioning that the node betweenness centrality in the backbone is given as that in the original network. Figures 7(e) and 7(f) give the evolution of the cumulative betweenness centrality distribution with different α. For both test graphs, $C(BC)$ starts from a very low value if BEHwII^H applies a very small value of α, which implies that those low-connected nodes will not be included in the backbones.

Therefore, we can conclude that values of α in the range $[0.4, 0.8]$ are optimal, in the sense that backbones extracted by BEHwII^H in this region have a large proportion of high-connective nodes and essential connections, and the stable stationary degree/weight distributions, compared with the original network. It is important to stress that BEHwII^H also includes the connections with the largest weight present in the network. This is because the heavy tail of the $P(w)$ distribution is mainly determined by relevant large-scale weight. This is clearly illustrated in Figures 7(c) and 7(d).

The Inside Filtering Rule. We further explore the critical factor that contributes to the success of BEHwII^H. As discussed

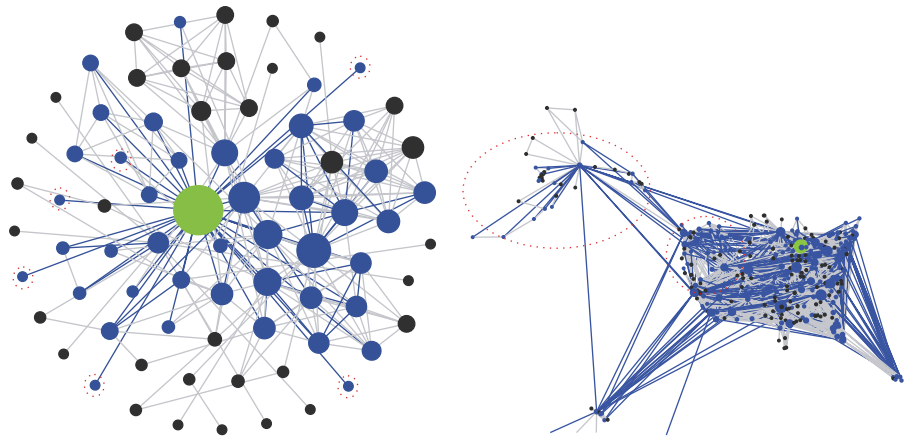


(a) USAir97: $C(k)$ (b) OClinks: $C(k)$

(c) USAir97: $P(w)$ (d) OClinks: $P(w)$

(e) USAir97: $C(BC)$ (f) OClinks: $C(BC)$

FIGURE 7: Impact of the threshold α.



(a) Lesmis (b) USAir97

Figure 8: Backbones extracted by BEHwII^{H*}.

in Section 3.1, BEHwII^{H} uses a strict filtering rule to absorb edges. Here, we relax the previous inside filtering rule, by imposing min instead of max, so that a connection is preserved whenever its intensity is significant for one of the nodes involved. In this loose BEHwII^{H} (denoted by BEHwII^{H*}), an edge e_{ij} is *preserved* in the backbone, if $\min(\alpha_{ij}, \alpha_{ji}) < \alpha$. We visualize the backbones of `Lesmis` and `USAir97` extracted by BEHwII^{H*} in Figure 8. For each test network, we set $\alpha = 0.5$ and $\%m_t = 0.25$. In the case of the `Lesmis` network, six outliers (highlighted by dashed circles) are extracted by BEHwII^{H*}, and it also fails to discover many essential connections. Obviously, its performance is worse than BEHwII^{H} by comparing Figures 8(a) and 5(a). BEHwII^{H*} has made progress in the case of `USAir97`, as most regions of USA have been covered in the extracted backbone as shown in Figure 8(b). However, it still includes many small airports in Alaska and the west coast of USA (highlighted in dashed ellipses) as DF does.

5. Related Work

In the literature, the existing backbone extraction methods can fall into two categories: the coarse graining based methods and the filter-based methods. The methods based on the coarse graining clump nodes sharing common attributes together in the same group/community and then consider the whole group as one single unit in the new networks. Some methods along this line include the box-covering technique [4], fractal skeleton [7], and traditional community detection techniques such as the Kernighan-Lin algorithm [11], latent space models [12], stochastic block models [13], and modularity optimization [14]. The differences between these methods ultimately come down to the precise definition of a community. However, there is often no clear statement on whether properties of the initial network are preserved in the network of groups.

The filter-based methods typically employ a bottom-up strategy to extract the backbone. They often start by defining a statistical property of a node or an edge, and then this property is used as a criterion to determine nodes/edges to be preserved or discarded. In this case, the observation scale is fixed and the representation that the network symbolizes is not changed. Instead, those elements, nodes, and edges, which carry relevant information about the network structure, are kept while the rest are discarded. An example of a well-known hierarchical topological filter is the k-core decomposition [16], with a filtering rule that acts on the connectivity of the nodes. In the case of weighted networks, two basic reduction techniques refer to the extraction of the minimum spanning tree [17] and the application of a global threshold [18] on the edge-weights, so that just those that beat the threshold are preserved, as real-world weighted networks that are usually with strong disorder heavy-tailed distributions of weight, which exerts nontrivial challenges to define the filtering property. Serrano et al. [8] addressed this challenge by introducing the disparity filter based on the null hypothesis.

In summary, although backbone extraction based on the coarse graining and filter models are extensively studied, they all need the knowledge of the entire network. Further study is still needed on finding a nice balance between the good performance and high efficiency. Our work attempts to fill this void by conducting backbone extraction based on an efficient BEHwII method.

6. Conclusion

In this work, we propose a backbone extraction heuristic with incomplete information (BEHwII) to find the backbone in a complex weighted network. First, a strict filtering rule is carefully designed to determine edges to be preserved or discarded. Second, we present a local search model to examine part of edges in an iterative way, which only relies on the local/incomplete knowledge rather than the global view of the network. Experimental results on four real-life networks demonstrate the advantage of BEHwII over the classic disparity filter method by either effectiveness or efficiency validity.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grants 61103229 and 71372188, the National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035, the National Key Technologies R&D Program of China under Grant 2013BAH16F01, the National Soft Science Research Program under Grant 2013GXS4B081, the Industry Projects in Jiangsu S&T Pillar Program under Grant BE2012185, and the Key/Surface Projects of Natural Science Research in Jiangsu Provincial Colleges and Universities under Grants 12KJA520001, 14KJA520001, and 14KJB520015.

References

[1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[2] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[3] H. Dong, Z. Wang, and H. Gao, "Robust H_∞ filtering for a class of nonlinear networked systems with multiple stochastic communication delays and packet dropouts," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 1957–1966, 2010.

[4] C. Song, S. Havlin, and H. A. Makse, "Self-similarity of complex networks," *Nature*, vol. 433, no. 7024, pp. 392–395, 2005.

[5] C. E. Hutchins and M. Benham-Hutchins, "Hiding in plain sight: criminal network analysis," *Computational and Mathematical Organization Theory*, vol. 16, no. 1, pp. 89–111, 2010.

[6] J. H. Choi, G. A. Barnett, and B.-S. Chon, "Comparing world city networks: a network analysis of Internet backbone and air transport intercity linkages," *Global Networks*, vol. 6, no. 1, pp. 81–99, 2006.

[7] K. I. Goh, G. Salvi, B. Kahng, and D. Kim, "Skeleton and fractal scaling in complex networks," *Physical Review Letters*, vol. 96, no. 1, Article ID 018701, 2006.

[8] M. Á. Serrano, M. Boguñá, and A. Vespignani, "Extracting the multiscale backbone of complex weighted networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 16, pp. 6483–6488, 2009.

[9] Z. Wu, L. A. Braunstein, S. Havlin, and H. E. Stanley, "Transport in weighted networks: partition into superhighways and roads," *Physical Review Letters*, vol. 96, no. 14, Article ID 148702, 2006.

[10] P. S. Dodds, D. J. Watts, and C. F. Sabel, "Information exchange and the robustness of organizational networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12516–12521, 2003.

[11] Y. Weihong, Y. Yuehui, and T. Guozhen, "Recursive Kernighan-Lin algorithm (RKL) scheme for cooperative road-side units in vehicular networks," in *Parallel Computational Fluid Dynamics*, vol. 405 of *Communications in Computer and Information Science*, pp. 321–331, Springer, Berlin, Germany, 2014.

[12] S. A. Stoev, G. Michailidis, and J. Vaughan, "On global modeling of backbone network traffic," in *Proceedings of the 29th Conference on Computer Communications (INFOCOM '10)*, pp. 1–5, San Diego, Calif, USA, March 2010.

[13] A. Kim and H. Krim, "Hierarchical stochastic modeling of SAR imagery for segmentation/compression," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 458–468, 1999.

[14] Z. Bu, C. Zhang, Z. Xia, and J. Wang, "A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network," *Knowledge-Based Systems*, vol. 50, pp. 246–259, 2013.

[15] D. Gfeller and P. de Los Rios, "Spectral coarse graining of complex networks," *Physical Review Letters*, vol. 99, no. 3, Article ID 038701, 2007.

[16] J. Chalupa, P. L. Leath, and G. R. Reich, "Bootstrap percolation on a Bethe lattice," *Journal of Physics C: Solid State Physics*, vol. 12, no. 1, pp. L31–L35, 1979.

[17] P. J. Macdonald, E. Almaas, and A.-L. Barabási, "Minimum spanning trees of weighted scale-free networks," *Europhysics Letters*, vol. 72, no. 2, pp. 308–314, 2005.

[18] V. M. Eguíluz, D. R. Chialvo, G. A. Cecchi, M. Baliki, and A. V. Apkarian, "Scale-free brain functional networks," *Physical Review Letters*, vol. 94, no. 1, Article ID 018102, 2005.

[19] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[20] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[21] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, vol. 37, Addison-Wesley, Reading, Pa, USA, 1993.

[22] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 11, pp. 3747–3752, 2004.

[23] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social Networks*, vol. 31, no. 2, pp. 155–163, 2009.

[24] S. R. Corman, T. Kuhn, R. D. Mcphee, and K. J. Dooley, "Studying complex discursive systems centering resonance analysis of communication," *Human Communication Research*, vol. 28, no. 2, pp. 157–206, 2002.