Community Structure in Large Complex Networks*

Liaoruo Wang
School of Electrical and Computer Engineering, Cornell University
Ithaca, NY 14853
lw335@cornell.edu

John E. Hopcroft
Department of Computer Science
Cornell University
Ithaca, NY 14853
jeh@cs.cornell.edu

ABSTRACT

In this paper, we establish the definition of community fundamentally different from what was commonly accepted in previous studies, where communities were typically assumed to be densely connected internally but sparsely connected to the rest of the network. A community should be considered as a densely connected subset in which the probability of an edge between two randomly-picked vertices is higher than average. Moreover, a community should also be well connected to the remaining network, that is, the number of edges connecting a community to the rest of the graph should be significant. In order to identify a well-defined community, we provide rigorous definitions of two relevant terms: "whiskers" and the "core". Whiskers correspond to subsets of vertices that are barely connected to the rest of the network, while the core exclusively contains the type of community we are interested in. We have proven that detecting whiskers, or equivalently, extracting the core, is an NP-complete problem for weighted graphs. Then, three heuristic algorithms are proposed for finding an approximate core and are evaluated for their performance on large networks, which reveals the common existence of the core structure in both random and real-world graphs. Further, well-defined communities can be extracted from the core using a number of techniques, and the experimental results not only justify our intuitive notion of community, but also demonstrate the existence of large-scale communities in various complex networks.

1. INTRODUCTION

Ever since people started to realize the importance of comprehending how interactions initiate and develop, the research on complex networks has attracted a great amount of attention. A substantial quantity of work has been devoted to the task of identifying and evaluating close-knit communities in large complex networks, most of which is based on the premise that it is a matter of common experience that communities exist in these networks [4]. In particular, as the Internet has become an indispensable part of our life, understanding community structure is not only crucial for studying real-world societies, but also helpful to improve the accuracy and reliability of predicting online behaviors, which may greatly benefit the quality and effectiveness of online services, such as search engines, recommendation systems, and so on.

A complex network is usually modeled as a graph in which vertices represent entities and edges represent interactions between pairs of entities. In previous studies, a community was often assumed to be a subset of vertices that are densely connected internally but sparsely connected to the rest of the network [2, 3, 4]. Accordingly, numerous measures have been proposed to capture this feature, out of which conductance has become one of the most widely adopted metrics for evaluating how community-like a subset of vertices is. Particularly, Leskovec et al. [4] conducted an extensive research on more than 100 large complex networks under the assumption that a community is more densely connected between its members than between its members and the remaining network. They carefully examined the relationship between conductance and community size, and discovered that the best community of the entire graph, i.e. the subset with the global minimum conductance, is usually a small set of vertices barely connected to the rest of the network by just a single edge.

However, it is our view that for real-world societies, communities are not only better connected than expected solely from chance, but are also well connected to the rest of the network. Actually, it is hard to imagine a small close-knit community, such as an academic department, with only one edge connecting it to the outside world. Empirically, a community displays a higher than average edge-vertex^2 ratio, which reflects the probability of an edge between two randomly-picked vertices, and it is also connected to the rest of the network via a significant number of edges, which is even possibly larger than the number of its internal edges, as depicted in Fig. 1.

Given a subset of vertices, an edge with only one endpoint inside the subset can be thought as a cut edge. A densely connected subset with a small number of cut edges, called a whisker, is not the type of community we are interested in. Since many previously-used measures simultaneously maximize internal connections and minimize external connections, leaving whiskers in the graph will interfere with the algorithms intended to extract the type of community we are interested in. Whiskers are peripheral rather than central, thus, the type of community we would like to identify is embedded in a special structure in which no whiskers exist, called the core. To get rid of the interference generated by whiskers, a community detection algorithm can be designed consisting of two steps: 1) identifying the core in which no whiskers exist, and 2) identifying communities in the core.

*This research was partially supported by the U.S. Air Force Office of Scientific Research under Grant FA9550-09-1-0675.

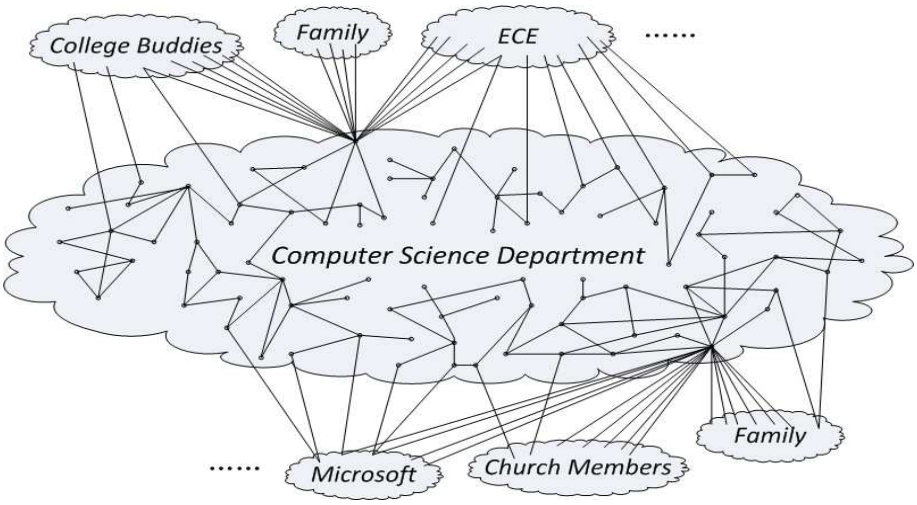


Figure 1: An example friendship network. Vertices typically have a significant number of cut edges.

Apparently, any subset of the core is connected to the rest of the graph by a moderate number of edges, and conductance can still be taken as a measure of community goodness. In this way, the best community is not only more densely connected than expected from chance but also well connected to the remaining network, which exactly corresponds to our intuitive notion of community.

We prove that extracting the exact core from a weighted graph is NP-complete, and then conjecture the unweighted version of this problem to be also NP-complete. It is not difficult to see that, generally, the exact core cannot be obtained by removing whiskers one by one, but removing whiskers in a certain way can lead to an approximate core. We develop three heuristic algorithms, all of which are capable of finding an approximate core. Their performance can be verified by the experimental results obtained from random graphs and real-world graphs. In addition, we also discover that some algorithms are only suitable for a certain kind of networks but not for others. Further, the algorithms can be justified by the community profile of the core, in contrast to that of the entire graph shown in [4], which plots the smallest possible conductances with respect to fixed community sizes. In various complex networks, the best communities have a relatively large conductance, which means the communities are densely connected internally while preserving a significant number of cut edges. Moreover, they also have a relatively large size, which demonstrates the existence of large-scale well-defined communities.

The rest of this paper is organized as follows. In Section 2, we introduce some necessary background and present definitions of whiskers and the core. Then, in Section 3, we prove the NP-completeness of finding the exact core in weighted graphs and propose three heuristic algorithms for finding an approximate core. In Section 4, we apply the algorithms to random graphs and real-world graphs to evaluate their performance and compare the experimental results. Finally, we conclude in Section 5 with comments on the problems considered and future work.

2. THE CORE STRUCTURE

In this section, we first review some previous research on community discovery in large complex networks. Then, we provide rigorous definitions of several terminologies related to whiskers and the core, which establish the theoretical foundation of this paper. Finally, we prove two preliminary lemmas to explore properties of whiskers and the core.

2.1 Background

Given an undirected graph $G = (V, E)$ with adjacency matrix A, the *conductance* of a subset of vertices $S \subseteq V$ is defined as

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\{D(S), D(S^c)\}}.$$

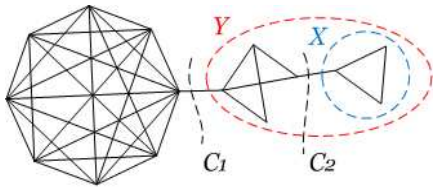
Here, S^c denotes the complement of S and

$$D(S) = \sum_{i \in S} \sum_{j \in V} A_{ij} = \sum_{i \in S} d(i),$$

where $d(i)$ denotes the degree of vertex i in the graph G [4]. Clearly, the conductance of S provides a measure for the quality of the corresponding cut, which divides the graph into two subsets S and S^c. Out of numerous density-based measures, conductance has been extensively employed for community detection, which intends to maximize internal connectivity and minimize external connectivity [2, 6].

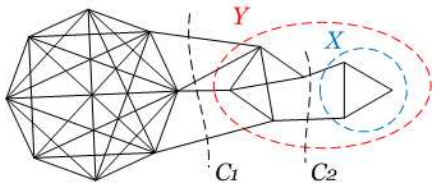
The concept of whiskers was informally introduced in [4] referring to weakly-connected subsets linked to the rest of the graph by just a single edge. Empirically, whiskers are peripheral and can be removed from the graph using the depth-first search to extract the giant biconnected component. Then, the union of whiskers is considered to form the periphery of the graph and the giant biconnected component is considered as the core. However, the biconnected component may still display a core-periphery structure with whiskers now referring to weakly-connected subsets linked to the rest of the graph by two edges, which inspires our generalized definitions of whiskers and the core.

In [4], a large number of real-world complex networks, such as friendship, citation, email and road networks, were thoroughly explored. Also, several approximation algorithms for community identification were implemented and evaluated. These algorithms typically return a whisker or a union of disjoint whiskers as the best community, thus, whiskers



X: a 1-whisker but not a maximal 1-whisker
Y: a 1-whisker and also a maximal 1-whisker
C1, C2: suitable cut

(a) k-whisker vs. maximal k-whisker



X: a whisker but not a maximal whisker
Y: a whisker and also a maximal whisker
C1, C2: suitable cut

(b) whisker vs. maximal whisker

Figure 2: Schematic illustration of Definition 2.1 through Definition 2.5.

are often interpreted as meaningful communities and are believed to have a significant influence on the community structure of the entire network. In addition, the network profile plot was also introduced to uncover the relationship between the lowest conductances and fixed community sizes, which usually achieves a global minimum at a small size scale of roughly 100 vertices.

A close-knit subset with just a single edge connecting it to the rest of the graph often corresponds to a low-conductance subset, which is more likely to be extracted as the optimal solution by the algorithms designed to minimize the conductance, and this may explain the reason why the best community is usually a whisker or a union of disjoint whiskers.

2.2 "Whiskers" and the "Core"

Let $G = (V, E)$ be an undirected graph with n vertices and m edges. A cut C is a collection of edges such that removing them from the graph G separates the vertex set V into two disjoint subsets S and S^c, where S^c denotes the complement of S and $C = \{(v, w) \in E \mid v \in S;\ w \in S^c\}$. Without loss of generality, we assume $|S| \leqslant |S^c|$ throughout this paper, where $|S|$ and $|S^c|$ denote the cardinality of sets S and S^c, respectively. Note that both S and S^c are not necessarily connected. Then, an edge $(v, w) \in C$ is called a *cut edge*, and intuitively, the cut size is defined to be the cardinality of the set C. Further, a cut is considered to be *suitable* if its removal divides the vertices into two disjoint subsets such that both have cardinality greater than or equal to the cut size.

Definition 2.1. *A cut of size k is a **suitable cut** if its removal from the graph partitions the vertex set into two disjoint subsets S and S^c, where $k \leqslant |S| \leqslant |S^c|$.*

Leskovec et al. [4] defined 1-whiskers to be maximal subgraphs that can be detached from the rest of the graph by removing a single edge, and they also use the term "whiskers" informally to refer to subsets of vertices barely connected to the rest of the graph. Whiskers are generally quite small compared to the whole graph while possessing a wide range of sizes and shapes. Moreover, they usually correspond to low-conductance sets that are more densely connected inside than connected to the outside. Hence, whiskers and unions of disjoint whiskers are believed to exert a significant effect on the community structure of real-world networks, since they are extracted and interpreted as communities by the conductance measure, which, out of numerous density-based measures, has been extensively used for detecting communities and evaluating their quality [2, 4, 6].

However, as clarified in Section 1, this type of community neither corresponds to our intuitive notion of community nor widely exist in real-world societies, where it is a matter of common observation that communities are not only densely connected inside but also well connected to the outside. Therefore, it is of major interest to remove whiskers from the graph in order to provide insight into the community structure of the network core. For this purpose, we rigorously define whiskers and the corresponding core structure where barely-connected subsets have been removed.

Definition 2.2. *Given an undirected graph $G = (V, E)$ with n vertices, a **k-whisker** is defined as a connected subgraph $G_w(k) = (V_w(k), E_w(k))$ linked to the rest of the graph by k edges, where $k \leqslant |V_w(k)| \leqslant n/2$.*

Definition 2.3. *Given an undirected graph $G = (V, E)$ with n vertices, a **maximal k-whisker** is defined as a maximal connected subgraph $G^*_w(k) = (V^*_w(k), E^*_w(k))$ linked to the rest of the graph by k edges, where $k \leqslant |V^*_w(k)| \leqslant n/2$.*

Small isolated components are frequently encountered in large complex networks, and they can simply be viewed as (maximal) 0-whiskers. Definition 2.2 and 2.3 are a direct extension of the definition of 1-whiskers given in [4]. Then, in a similar way, the definitions of whiskers and maximal whiskers can be formulated independent of the value of k referring to weakly-connected subsets attached to the remaining graph via a small number of edges.

Definition 2.4. *Given an undirected graph $G = (V, E)$ with n vertices, a **whisker** is defined as a connected subgraph $G_w = (V_w, E_w)$ linked to the rest of the graph by a suitable cut, where $|V_w| \leqslant n/2$.*

Definition 2.5. *Given an undirected graph $G = (V, E)$ with n vertices, a **maximal whisker** is defined as a maximal connected subgraph $G^*_w = (V^*_w, E^*_w)$ linked to the rest of the graph by a suitable cut, where $|V^*_w| \leqslant n/2$.*

See Fig. 2 for a detailed illustration of Definition 2.1 through Definition 2.5. A maximal whisker is obviously a whisker, but a whisker is not necessarily a maximal whisker, since it can be contained in a larger whisker. Besides, a 0-whisker is also a maximal whisker by Definition 2.5.

As discussed above, maximal whiskers, although argued by some to be community-like, are not what we are interested in here. Therefore, we define the core as the remaining structure after removing the union of all maximal whiskers

from the graph. Meaningful communities can be further extracted from the core using a variety of algorithms, which, unlike whiskers, are not only better connected than expected from chance but also well connected to the rest of the graph.

DEFINITION 2.6. *The* **core** *is a connected subgraph that is the complement of the union of all maximal whiskers.*

Clearly, there does not exist any suitable cut in the core subgraph. Before we move on to Section 3 to design and implement algorithms for finding the core structure and its underlying communities, we first examine some properties of whiskers. If all maximal whiskers are disjoint in the graph, it is straightforward that we can remove these disjoint whiskers one by one until we obtain the core. However, whiskers may overlap with each other, and unfortunately, their union is often no longer a whisker. In fact, a number of counterexamples can be constructed to justify this statement, and we conclude the following lemma:

LEMMA 2.7. *Let G be an undirected graph with two overlapping maximal whiskers S and T. The subgraph $S \cup T$ is not necessarily a whisker.*

PROOF. As shown in Fig. 3, for instance, $S = X \cup Y$ is a maximal whisker with 22 vertices and 21 outgoing edges. Similarly, $T = Y \cup Z$ is also a maximal whisker with 20 vertices and 19 outgoing edges. However, there are a total of 25 vertices in the set $X \cup Y \cup Z$ and 26 outgoing edges that connect this union to the rest of the graph, thus $S \cup T$ is not a whisker. □

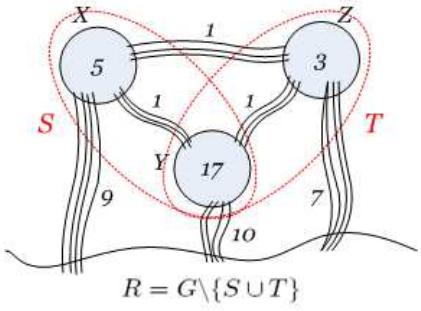


Figure 3: Two overlapping maximal whiskers whose union is not a whisker. Each circled integer denotes the number of vertices in the corresponding set.

In general, there are two reasons why a subset of vertices is not a whisker: 1) it contains more than half of the vertices, and 2) the number of edges connecting it to the rest of the graph is strictly greater than its cardinality. Thus, the union of two disjoint whiskers is still a whisker if and only if it is no larger than its complement. In addition, the union of two overlapping maximal whiskers is not a maximal whisker, since a maximal whisker cannot have any other maximal whisker as its subset. Based on Lemma 2.7, there is another observation we can make about whiskers:

LEMMA 2.8. *Let G be an undirected graph with n vertices and two overlapping whiskers S and T, where the number of vertices in the subgraph $S \cup T$ is no more than $n/2$. If $S \cup T$ is not a whisker, then the subgraph $S \cap T$ must be a whisker.*

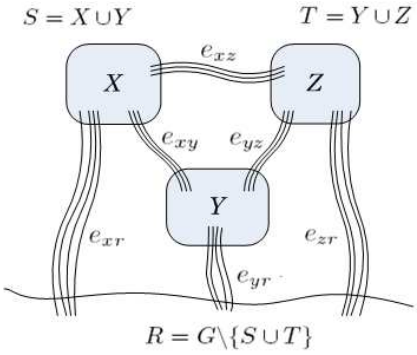


Figure 4: Two overlapping whiskers.

PROOF. Assuming that the subgraph $S \cup T$ is not a whisker, write $S = X \cup Y$ and $T = Y \cup Z$ where $S \cap T = Y$, as shown in Fig. 4.
Then, it follows that

$$e_{xr} + e_{xz} + e_{yr} + e_{yz} \leqslant v_x + v_y \tag{1a}$$
$$e_{yr} + e_{xy} + e_{zr} + e_{xz} \leqslant v_y + v_z \tag{1b}$$
$$e_{xr} + e_{yr} + e_{zr} > v_x + v_y + v_z \tag{1c}$$

where v_x, v_y, and v_z denote the number of vertices in the sets X, Y, and Z, respectively. Adding Equation (1a) and (1b), we have that

$$\begin{aligned} e_{xr} + 2e_{yr} + e_{zr} + e_{xy} + e_{yz} + 2e_{xz} &\leqslant v_x + 2v_y + v_z \\ &< e_{xr} + e_{yr} + e_{zr} + v_y. \end{aligned}$$

Thus,

$$e_{yr} + e_{xy} + e_{yz} + 2e_{xz} < v_y. \tag{1d}$$

Since e_{xz} is non-negative as the number of edges between the sets X and Z, by Equation (1d),

$$e_{yr} + e_{xy} + e_{yz} < v_y,$$

and the subgraph $Y = S \cap T$ is clearly a whisker. □

3. METHODOLOGY

In this section, we discuss the approach for efficiently identifying the core of a given graph. Armed with the definitions provided in Section 2.2, we prove in Section 3.1 that detecting whiskers in a weighted undirected graph is NP-complete and thus computationally intractable unless P=NP. This indicates that there is no feasible algorithm for finding the exact core, which is equivalent to finding the union of all maximal whiskers. Then, in Section 3.2, we propose three heuristic algorithms for finding an approximate core, whose performance will be experimentally justified in Section 4.

3.1 NP-Completeness

Define NAE-3-SAT as the problem of determining whether there exists a truth assignment for a 3-CNF Boolean formula such that each clause has at least one true literal and at least one false literal (i.e. literals in each clause are not all equal). Then, we have the following well-known theorem:

THEOREM 3.1. NAE-3-SAT *is* NP*-complete* [5].

Now, define WHISKER as the problem of determining whether there exists a whisker in a given weighted undirected graph.

We will formally prove that WHISKER is also an NP-complete problem by constructing a polynomial-time reduction from NAE-3-SAT.

THEOREM 3.2. WHISKER *is* NP*-complete.*

PROOF. Given an instance of the WHISKER problem, we can guess a solution and verify in linear time whether it is indeed a whisker, thus WHISKER $\in$ NP.

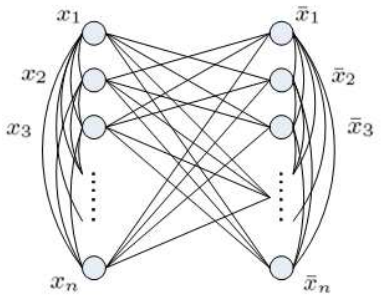


Figure 5: The weighted graph G.

Consider a weighted undirected graph G as depicted in Fig. 5. A total of $2n$ vertices are arranged in two columns of n vertices each, which correspond to literals $\{x_i, \bar{x}_i | 1 \leqslant i \leqslant n\}$. Each vertex is connected by an edge to every other vertex except its complement in the other column, and each edge is assigned weight $1/(n-1)$. Note in particular that the size of any cut in G has been generalized to the weighted sum of the cut edges. Clearly, this graph can be constructed in polynomial time.

Pick one vertex from each row and the resulting subgraph has n vertices and $n(n-1)$ cut edges. Since

$$n(n-1) \cdot \frac{1}{n-1} = n,$$

this subgraph is actually a whisker by Definition 2.4. Hence, there are 2^n such whiskers and we claim that no more whiskers can be found in the graph. Suppose that there is a whisker that has $2k+j$ vertices consisting of both vertices from k rows and one of the two vertices from j rows. We require that $2k+j \leqslant n$ since a whisker cannot contain more than half of the vertices. Then, the whisker gives a cut size of

$$\frac{1}{n-1}\left[2k(2n-2k-j)+j(2n-2k-j-1)\right].$$

According to the hypothesis,

$$\frac{1}{n-1}\left[(2k+j)(2n-2k-j)-j\right] \leqslant 2k+j,$$

and it follows that

$$(2k+j)(n-2k-j+1) \leqslant j. \tag{2}$$

For $k=0$, the inequality holds only if $j=n$, and this simply corresponds to one of the 2^n whiskers we previously found. For $k \geqslant 1$ and $j \leqslant n-2k$, the inequality cannot hold since $2k+j \geqslant j+2$ and $n-2k-j+1 \geqslant 1$. Thus, no other whiskers exist in the graph and these 2^n whiskers are also maximal whiskers. Note that the union of any two of these whiskers is no longer a whisker, since it contains more than n vertices.

Next, consider a given 3-CNF Boolean formula with c clauses and n variables. For each i $(1 \leqslant i \leqslant c)$, we can construct a unit-weighted undirected graph G_i for the ith clause with $2n$ vertices arranged in two columns of n vertices each, which represent the n variables and their negations, respectively. Such a graph G_i is complete except that there are no edges between pairs of literals in the ith clause, as shown in Fig. 6.

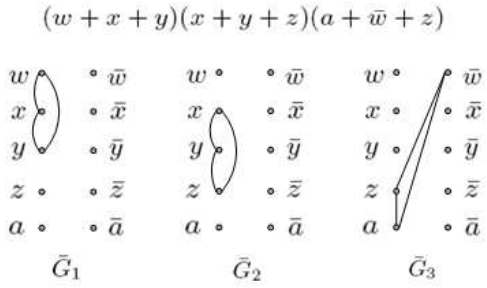


Figure 6: The unit-weighted graphs $G_1, G_2, \cdots, G_c$ corresponding to the example clauses. For graphical simplicity, the complement of each graph is drawn instead of the original graph.

Then, the graphs $G_1, G_2, \cdots, G_c$ can be combined into a weighted graph G' by accumulating the weight of each edge while preserving the label of each vertex, as shown in Fig. 7.

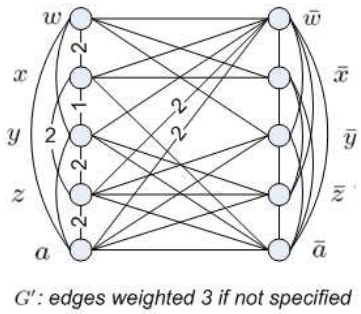


Figure 7: The weighted graph G' obtained by combining G_1, G_2, G_3 shown in Fig. 6. The graph G' is complete but not all edges are displayed. The edges are weighted by 3 except those specified above.

Reduce the edge weights of G by a small amount to $1/(n-\varepsilon)$, where $0 \ll \varepsilon < 1$. Then, the 2^n whiskers have a slight excess of vertices but no new whiskers are formed, which allows more edges to be added. Scale the edge weights of G' by a small amount $\delta > 0$, where $cn^2\delta \ll 1$. Then, G and G' can be merged into a weighted graph G^*, replacing the vertex labels of G by those of G' and summing their corresponding edge weights. Clearly, the whiskers in G^*, like those in G, still come from selecting one vertex from each row.

For any truth assignment, rearrange the graphs with the two columns corresponding to true literals and false literals, respectively. If there is a not-all-equal truth assignment for the Boolean formula, then each clause must have one true literal and one false literal, which indicates that the literals of each clause cannot lie within the same column of the rearranged graph. Clearly, for each i, there are n^2-2 edges

connecting the two columns of G_i with no edges between the literals of the ith clause. Thus, the weighted sum of the edges connecting the two columns of G' is given by $cn^2 - 2c$.

Selecting the true literal from each row, we have a subset of n vertices that is linked to the rest of G' by $cn^2 - 2c$ edges. For this subset to be a whisker after merging G and G', we require that

$$\frac{1}{n-\varepsilon} n(n-1) + \delta\left(cn^2 - 2c\right) \leqslant n. \tag{3a}$$

In contrast, if there is no such truth assignment for the Boolean formula, then at least one clause has its literals located within the same column of the rearranged graph. Clearly, there are $n^2 - 2$ edges connecting the two columns for at most $c-1$ of the c graphs, and n^2 edges connecting the two columns for at least one of them. Thus, the weighted sum of the edges connecting the two columns of G' is at least $(c-1)(n^2-2)+n^2 = cn^2-2c+2$. Selecting one literal from each row, we have a subset of n vertices that is linked to the rest of G' by at least cn^2-2c+2 edges. For this subset not to be a whisker after merging G and G', we require that

$$\frac{1}{n-\varepsilon} n(n-1) + \delta\left(cn^2 - 2c + 2\right) > n. \tag{3b}$$

Combining Equation (3a) and (3b), it follows that

$$\frac{n(1-\varepsilon)}{(n-\varepsilon)\left(cn^2-2c+2\right)} < \delta \leqslant \frac{n(1-\varepsilon)}{(n-\varepsilon)\left(cn^2-2c\right)}. \tag{4}$$

With ε and δ satisfying Equation (4) for the given c and n, the true literals of a not-all-equal assignment for the formula correspond to the vertices of a whisker in G^*, and the vertices of a whisker in G^* also correspond to the true literals of a not-all-equal assignment for the formula. Therefore, we have established a one-to-one correspondence between not-all-equal truth assignments and whiskers, that is, a weighted graph can be constructed for a given 3-CNF Boolean formula such that whiskers can be found in the graph if and only if the formula is not-all-equal satisfiable. Clearly, NAE-3-SAT reduces to WHISKER in polynomial time, thus, WHISKER is NP-complete. □

We then conjecture that detecting whiskers in an unweighted graph is also an NP-complete problem.

3.2 Heuristic Algorithms

An intuitive approach to identifying the core is simply to remove maximal whiskers one by one until no more whiskers exist. However, the following claim characterizes the non-exactness and non-uniqueness of this method, which indicate the generic difficulties associated with any algorithm using this approach to find the core structure.

CLAIM 3.3. *Removing maximal whiskers one by one leads to different subgraphs approximate to the exact core, depending on the order in which whiskers are removed.*

PROOF. Here, we can still take Fig. 3 as an example. Assume that sets S and T are both maximal whiskers and that they do not intersect with other maximal whiskers. If the set S is first removed, we will be left with the set Z of 3 vertices and 7 outgoing edges, which is apparently not a (maximal) whisker. However, if the set T is first removed instead, we will be left with the set X of 5 vertices and 9 outgoing edges, which is not a (maximal) whisker either. In this case, different sets of vertices remain as part of the ultimate subgraph, neither of which belongs to the exact core. Therefore, the approximate core subgraph depends rather crucially on the order in which we remove these maximal whiskers from the graph, which means that it is not necessarily unique. □

The NP-completeness of identifying the exact core in weighted graphs has been proved in Section 3.1. We conjecture that identifying the exact core in unweighted graphs is also an NP-complete problem. Now, we present three heuristic algorithms for finding an approximate core, whose performance on random and real-world graphs will be experimentally demonstrated in Section 4.

Algorithm 1 (brute-force search)
For each ordered pair of vertices, find its minimum cut and remove the smaller component if the cut is suitable.

Algorithm 2
Extract the giant component and then the giant biconnected component. Replace all degree-two vertices by a single edge and then test the existence of suitable cuts.

Algorithm 3 (flow-based algorithm)
For a given threshold value λ, find the largest subgraph with the maximum edge-vertex ratio exceeding λ. Then, test the existence of suitable cuts.

There is no particular order in which whiskers are removed by Algorithm 1. According to Claim 3.3, larger maximal whiskers could be destroyed and the resulting graph is not necessarily unique, depending rather crucially on the order in which Algorithm 1 removes whiskers. Since a series of degree-two vertices could result in a whisker, Algorithm 2 contracts all degree-two vertices after obtaining the giant biconnected component. Although Algorithm 2 offers a better run-time performance compared to Algorithm 1, it actually encounters the same difficulties as Algorithm 1 does. The three algorithms are all capable of finding an approximate core, but we will focus on Algorithm 2 and Algorithm 3 since they require shorter running time. Empirically, Algorithm 2 works better for sparse networks, while Algorithm 3 works better for dense ones.

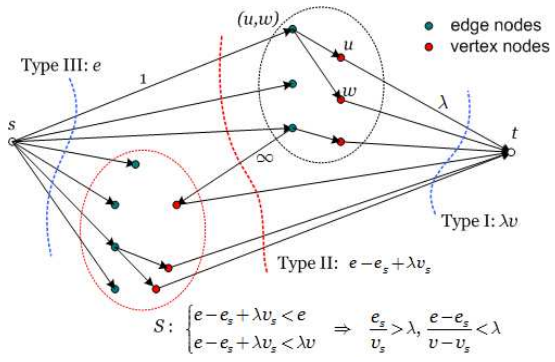
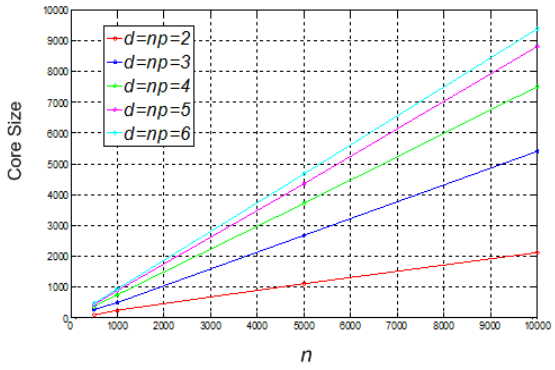
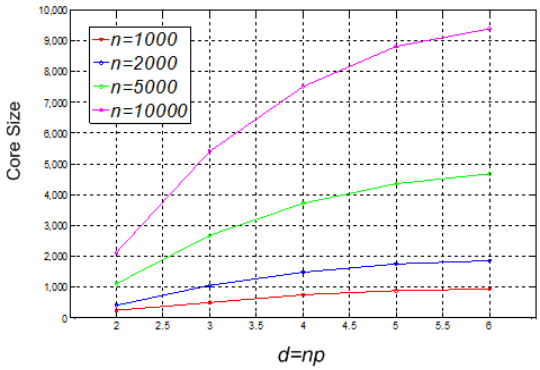


Figure 8: Algorithm 3 (flow-based algorithm).

Algorithm 3 is also known as a flow-based algorithm, where the original graph is converted into a weighted directed graph. For the new graph, we create two designated nodes as source and sink, plus a vertex/edge node corresponding to each vertex/edge of the original graph. A directed edge connects the



(a) size of the core as a function of n for fixed d



(b) size of the core as a function of d for fixed n

Figure 9: Random Graphs

source to every edge node with capacity 1 and connects every vertex node to the sink with capacity λ. Also, a directed edge of infinite capacity connects every edge node to each of the two vertex nodes corresponding to the two vertices adjacent to that edge. As shown in Fig. 8, a maximum flow algorithm will return three types of minimum cut set: 1) all the edges going into the sink, 2) some edges coming from the source and some edges going into the sink, and 3) all the edges coming from the source. Note that the directed edges of infinite capacity are opposite from others such that they are never counted as cut edges.

Assume that the original graph has v vertices and e edges. By adjusting the value of λ, we would like to obtain the second type of minimum cut set, where the capacity of minimum cut is given by $e - e_s + \lambda v_s$. Here, e_s and v_s denote the number of edge nodes and vertex nodes in the set S, respectively. Hence, S contains a subset of edge nodes and vertex nodes that satisfy the conditions

$$\frac{e_s}{v_s} > \lambda \qquad \text{and} \qquad \frac{e - e_s}{v - v_s} < \lambda.$$

Clearly, S corresponds to a subgraph of the original graph with the maximum edge-vertex ratio exceeding λ. Note that this algorithm always returns the largest subgraph meeting the above requirements.

4. EXPERIMENTAL RESULTS

4.1 Random Graphs

A random graph $G(n, p)$ can be obtained by starting with a set of n vertices and adding (undirected) edges between them independently with probability $p \in (0, 1)$. Although a random graph does not display any community structure, we can still identify its core using the above algorithms. When p is relatively small, $G(n, p)$ is sparse with low edge-vertex ratio, where Algorithm 3 fails to find an approximate core. In this case, Algorithm 2 can positively identify an approximate core. When p is close to 1, both algorithms are successful in finding an approximate core. As illustrated in Fig. 9, the size of the core of $G(n, p)$ grows linearly with $d = np$ for fixed n and logarithmically with n for fixed d. In addition, we observe the existence of phase transition at $p = 1/n$, above which the core emerges with high probability and below which it emerges with extremely low probability.

We conjecture that every $G(n, p)$ with $p > 1/n$ displays the core structure with high probability. For any fixed (large) n, $p = 1/n$ is the threshold for phase transition at which the core structure emerges. The probability and the average size of the core both increase as p grows. For any fixed p, the average size of the core increases as n grows, but the probability of the core remains the same.

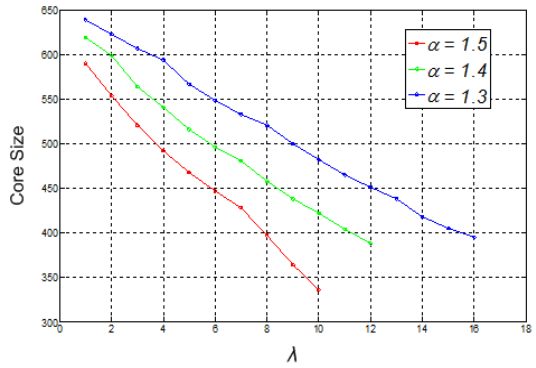
4.2 Real-World Graphs

4.2.1 Textual Graph

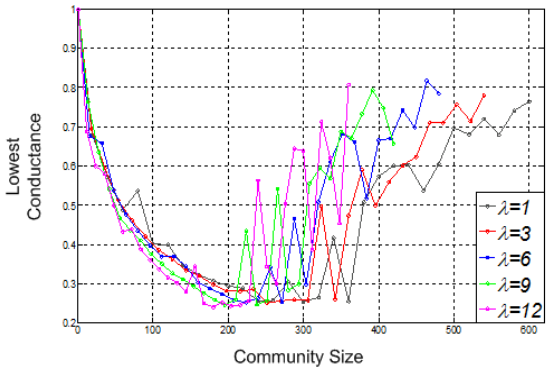
A textual graph consists of vertices representing words and edges representing semantic correlations, which contains information about research topics and areas of interest. We crawl more than 10,000 scientific papers of the KDD conference from 1992 to 2003 and collect the words of each abstract. A series of pre-processing steps are carried out to simplify the data, which include word stemming, stop-word filtering, and occurrence rate thresholding. Word stemming reduces inflected or derived words to their base form and combines multiple entries of the same word in different tenses. Stop-word filtering removes extremely common but meaningless words, such as and, can, the, will, etc. Occurrence rate thresholding removes extremely rare words occurring in only a small number of abstracts, which exert a trivial effect on the overall community structure.

Pointwise mutual information or log-likelihood ratio can be applied to determine whether there is an edge between each pair of vertices of the textual graph. In this section, we will only discuss the first approach. Pointwise mutual information quantifies the semantic correlation between two words, and we may choose a critical value α above which a strong correlation can be expected. In other words, if the mutual information of two words exceeds α, then an edge exists between them, which indicates a high probability for the two words to occur together. Otherwise, no edge exists between them, which indicates a low probability for the two words to occur together. For a pair of words (i, j) and the threshold value α, there exists an edge between vertex i and vertex j if

$$\log \frac{P(i, j)}{P(i)P(j)} > \alpha,$$

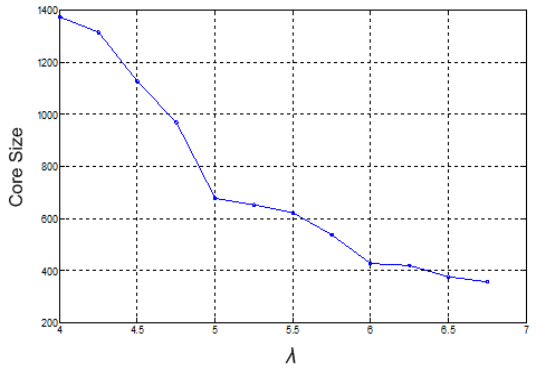


(a) size of the core as a function of λ

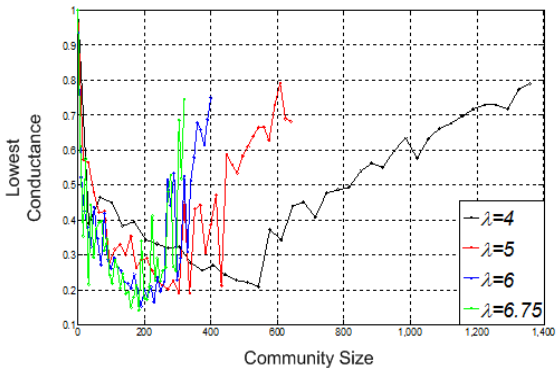


(b) community profile of the core ($\alpha = 1.4$)

Figure 10: Textual Graph



(a) size of the core as a function of λ



(b) community profile of the core

Figure 11: Co-authorship Graph

where $P(i)$ and $P(j)$ are the occurrence rate of i and j, respectively, and $P(i, j)$ is the probability of i and j occurring in the same abstract.

For example, the textual graph has 685 vertices and 6,432 edges when $\alpha = 1.4$. Both Algorithm 2 and Algorithm 3 are successful in identifying an approximate core, in which no whiskers exist. In particular, the core returned by Algorithm 2 is almost identical to that returned by Algorithm 3 when λ is relatively small. Higher values of λ will result in a smaller core, and intuitively, higher values of α will result in a graph with less edges and thus a smaller core, as verified in Fig. 10(a).

After the approximate core has been extracted from the graph, a simulated annealing algorithm can be performed on the core for finding a subset of a given size with the lowest conductance. As shown in Fig. 10(b), the best community of the textual graph possesses a quite large conductance around 0.3, which means the best community has only as many internal edges as cut edges. This exactly corresponds to our intuitive notion that a community should have a significant number of edges connecting it to the rest of the graph. Clearly, the community profile of the core is rather different from what was obtained in [4]. Recall that the best community of most networks examined in [4] displayed an extremely small conductance, typically at the order of 10^{-2}, which means the best community has almost 50 times as many internal edges as cut edges. Moreover, the best community of the textual graph is of size roughly 350 for $\alpha = 1.4$ and $\lambda = 1$, containing more than half of the vertices, which demonstrates the existence of large-scale well-defined communities. As expected, the best community extracted from the textual graph usually specifies a category of research topics or a flow of ideas, and Algorithm 3 is believed to be particularly useful when collaborative filtering is employed to improve the quality of search results.

4.2.2 *Co-authorship Graph*

A co-authorship graph reflects the common interests among researchers working in diverse fields, which contains information about authors' reputation and levels of activity. We collect more than 10,000 scientific papers of the KDD conference from 1992 to 2003 and refine the authors' information [1]. Different from the textual graph discussed in Section 4.2.1, the co-authorship graph is deterministic with 7,943 vertices and 20,488 edges, where each vertex represents an author and each edge represents a co-authorship. Here, Algorithm 2 is not successful in finding an approximate core by pulling out the giant biconnected component and contracting degree-two vertices. In contrast, Algorithm 3 is

able to identify an approximate core, and its size decreases as the threshold value λ increases, as shown in Fig. 11(a).

As depicted in Fig. 11(b), the community profile of the core of the co-authorship graph is rather different from what was obtained in [4]. Recall that the best community of most networks examined in [4] displayed an extremely small conductance, typically at the order of 10^{-2}, which means the best community has almost 50 times as many internal edges as cut edges. Here, the best community of the co-authorship graph possesses a quite large conductance around 0.2, which means the best community has only twice as many internal edges as cut edges. This, again, corresponds to our intuitive notion that a community should have a moderate number of edges connecting it to the rest of the graph. Moreover, the best community of the co-authorship graph is of size roughly 500 for $\lambda = 4$, containing more than a third of the vertices, which again demonstrates the existence of large-scale well-defined communities.

5. SUMMARY AND FUTURE WORK

We investigated large real-world complex networks and proposed an innovative definition of community as opposed to what was generally assumed in previous studies, where communities were thought to be better connected internally than connected with the rest of the network. In fact, a community is more densely connected internally than expected solely from chance, but it is also connected to the rest of the network by a significant number of edges. Further, we defined two auxiliary terms: whiskers and the core. Whiskers were often interpreted as communities, but they are not the type of community we are interested in here. In contrast, the core exclusively contains the type of community we would like to identify.

Armed with these definitions, we designed a community detection algorithm consisting of two steps: 1) identifying the core in which no whiskers exist, and 2) identifying communities within the core. However, extracting the exact core is rigorously proved to be NP-complete for weighted graphs, and we also conjecture the NP-completeness of this problem for the unweighted case. The three heuristic algorithms demonstrate their capability of finding an approximate core, and a simulated annealing algorithm is performed on the approximate core to find its best community, i.e. the subset with the lowest conductance, for a given community size. As expected, the network community profile of the core justifies our definition of community and shows the existence of large-scale well-defined communities in various real-world complex networks.

Overlapping communities exist more commonly than disjoint ones in real-world networks, but the community definition provided here does not include overlapping situations explicitly. Thus, a rigorous definition for overlapping communities is required, based on which we can design community detection algorithms and evaluate their performance. In this case, we still have the same notion that a community should not only be more densely connected than expected from randomness, but also be well connected to the rest of the network. Personal communities are another interest of our future research. We expect to find the union of all communities containing a particular vertex, and by taking the intersection of personal communities of different vertices, we can thoroughly understand the network structure from a community perspective.

6. REFERENCES

[1] arXiv data, 1992–2003. `http://www.cs.cornell.edu/projects/kddcup`.

[2] Gaertler, M. Clustering. In *Network Analysis: Methodological Foundations* (2005), U. Brandes and T. Erlebach, Eds., Springer, pp. 178–215.

[3] Girvan, M., and Newman, M. E. J. Community structure in social and biological networks. In *Proc. Natl. Acad. Sci. USA* (2002), pp. 7821–7826.

[4] Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. Community structure in large networks: natural cluster size and the absence of large well-defined clusters. Tech. Rep. arXiv:0810.1355, 2008.

[5] Schaefer, T. J. The complexity of satisfiability problems. In *Proc. 10th Ann. ACM Symp. on Theory of Computing* (1978), Association for Computing Machinery, pp. 216–226.

[6] Schaeffer, S. E. Graph clustering. *Computer Science Review 1*, 1 (2007), 27–64.