Tree decompositions and social graphs

Aaron B. Adcock, Blair D. Sullivan & Michael W. Mahoney

TREE DECOMPOSITIONS AND SOCIAL GRAPHS

Aaron B. Adcock,[1,*] Blair D. Sullivan,[2] and Michael W. Mahoney[3]

[1]*Department of Electrical Engineering, Stanford University, Stanford, California, USA*

[2]*Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA*

[3]*International Computer Science Institute and Department of Statistics, University of California at Berkeley, Berkeley, California, USA*

Abstract *Recent work has established that large informatics graphs such as social and information networks have a nontrivial treelike structure when viewed at moderate-size scales. Here, we present results from the first detailed empirical evaluation of the use of tree decomposition (TD) heuristics for structure identification and extraction in social graphs. Although TDs have historically been used in structural graph theory and scientific computing, we show that—even with existing TD heuristics developed for those very different areas—TD methods can identify interesting structures in a wide range of realistic informatics graphs. Our main contributions are the following: we show that TD methods can identify structures that correlate strongly with the core-periphery structure of realistic networks, even when using simple greedy heuristics; we show that the peripheral bags of these TDs correlate well with low-conductance communities (when they exist) found by using local spectral computations; and we show that several types of large-scale "ground-truth" communities, defined by demographic metadata on the nodes of the network, are well localized in the large-scale and/or peripheral structures of the TDs. Our other main contributions are as follows: we provide detailed empirical results for TD heuristics on toy and synthetic networks in order to establish a baseline to understand better the behavior of the heuristics on more complex real-world networks, and we prove a theorem providing formal justification for the intuition that the only two impediments to low-distortion hyperbolic embedding are high treewidth and long geodesic cycles. Our results suggest future directions for improved TD heuristics that are more appropriate for realistic social graphs.*

1. INTRODUCTION

Understanding the properties of realistic informatics graphs, such as large social and information networks, and developing algorithmic and statistical tools to analyze such graphs is of continuing interest, and recent work has focused on identifying and exploiting what may be termed treelike structure in these real-world graphs. Because an undirected graph is a tree if any two vertices are connected by exactly one path, or equivalently if

the graph is connected but has no cycles, real-world graphs are clearly not trees in any naïve sense of the word. For example, realistic social graphs have nonzero clustering coefficient, indicating an abundance of cycles of length three. There are, however, more sophisticated notions that can be used to characterize the manner in which a graph may be viewed as treelike. These are of interest because, e.g., graphs that are trees have many beneficial algorithmic and statistical properties, and the hope is that graphs that are treelike inherit some of these properties. In particular, δ-hyperbolicity is a notion from geometric group theory that quantifies a way in which a graph is treelike in terms of its distance or metric properties. Alternatively, tree decompositions (TDs) are tools from structural graph theory that quantify a way in which a graph is treelike in terms of its cut or partitioning properties.

Although TDs and δ-hyperbolicity capture very different ways in which a general graph can be treelike, recent empirical work (described in more detail below) has shown interesting connections between them. In particular, for realistic social and information networks, these two notions of treelikeness capture very similar structural properties, at least when the graphs are viewed at large-size scales; and this structure is closely related to what may be termed the *nested core-periphery* or k-core structure of these networks. Recent work has also shown that computing δ-hyperbolicity exactly is extremely expensive, that hyperbolicity is quite brittle and difficult to work with for realistic social graphs, and that common methods to approximate δ provide only a very rough guide to its extremal value and associated graph properties. Motivated by this, as well as by the large body of work in linear algebra and scientific computing on practical methods for computing TDs, in this article we present results from the first detailed empirical evaluation of the use of TD heuristics for structure identification and extraction in social graphs.

A TD (defined more precisely below) is a specialized mapping of an arbitrary input graph G into a tree H, where the nodes of H (called bags) consist of overlapping subsets of vertices of G. Quantities such as the treewidth—the size of the bag in H that contains the largest number of vertices from G—can be used to characterize how treelike G is. A single bag that contains every vertex from G is a legitimate but trivial TD (because the width is as large as possible for a graph of the given size). Thus, one usually focuses on finding "better" TDs, where better typically means minimizing the width. The problem of finding the treewidth of G and of finding an optimal TD of G are both NP-hard, and thus, most effort has focused on developing heuristics, e.g., by constructing the TD iteratively by choosing greedily vertices of G that minimize quantities such as the degree or fill. Because we are interested in applying TDs on realistic graphs, it is these heuristics (to be described in more detail following) that we will use in this article.

Our goals are to describe the behavior of TD heuristics on real-world and synthetic social graphs and to use these TD tools to identify and extract useful structure from these graphs. In particular, (in Section 6) we show the following. We first show (in Section 6.1) that TD methods can identify large-scale structures in realistic networks that correlate strongly with the recently described core-periphery structure of these networks, even when using simple greedy TD heuristics. We do this by relating the global "core-periphery" structure of these networks, as captured using the k-core decomposition, with what we call the "central-perimeter" structure, which is a measure of the centrality or eccentricity of each bag in the TD. We also describe how small-scale structures such as the internal bag structure of TDs of these networks reflects—depending on the density and other properties of these networks—their clustering coefficient and other related clustering properties of the original networks. We next show (in Section 6.2) that the peripheral bags of these

TDs correlate well with low-conductance communities/clusters (when they exist) found by using local spectral computations, in the sense that these low-conductance (i.e., good-conductance) communities/clusters occupy a small number of peripheral bags in the TDs. In particular, this shows that in graphs for which the so-called Network Community Profile (NCP) Plot is upward-sloping (as, e.g., described in [77], and indicating the presence of good small and absence of good large clusters), the small-scale "dips" in the NCP are localized in clusters that are on a peripheral branch in the TD. We finally consider (in Section 6.3) how several types of large-scale *ground-truth* communities/clusters, as defined by demographic metadata on the nodes of the network (and that are not good-conductance clusters), are localized in the TDs. In particular, we look at two social network graphs consisting of friendship edges among students at a university; we use metadata associated with graduation year and residence hall information, and we show that clusters defined by these metadata are well localized in the large-scale central and/or small-scale peripheral structures of the TDs.

A significant challenge in applying existing TD heuristics—which have been developed for very different applications in scientific computing and numerical linear algebra—is that it can be difficult to determine whether one is observing a "real" property of the networks or an artifact of the particular TD heuristic that was used to examine the network. Thus, to establish a baseline and to determine their behavior in idealized settings, we have first applied several existing TD heuristics to a range of toy and synthetic data. (See Section 4 and Section 5, respectively.) The toy data consist of a binary tree, a lattice, a cycle, a clique, and a dense random graph, i.e., graphs for which optimal TDs are known. The synthetic data consists of Erdős–Rényi and power law random graph models, which help us understand the effect of noise/randomness on the TDs. (Other random graph models exhibit similar properties when their parameters are set to correspondingly sparse values.) For these graphs, we place a particular emphasis on the properties of the TDs as the density parameters (i.e., the connection probability for the Erdős–Rényi graphs and the power law parameter for the power law graphs) are varied from very sparse to extremely sparse, and we are interested in how this relates to the large-scale core-periphery structure.

Our detailed empirical results for TD heuristics on toy and synthetic networks are important for understanding the behavior of these heuristics on more complex real-world networks; but our results on synthetic and real-world networks also suggest future directions for the development of TD heuristics that are more appropriate for social graph data. Existing TD heuristics focus on producing minimum-width TDs, which are of interest in more traditional graph theory and linear algebra applications, but they are not well optimized for finding structures of interest in social graphs. Although it is beyond the scope of this article, the development of TD heuristics that are more appropriate for social graph applications (e.g., understanding how the bag structure of those TDs relates to the output of recently developed local spectral methods that find good small clusters in large networks) is an important question raised by our results.

The remainder of this article is organized as follows. In Section 2, we present definitions from graph theory, a detailed discussion of tree decompositions and the algorithms for their construction, and a brief discussion of other prior related work. Section 3 details the datasets we make use of throughout the paper. The subsequent four sections provide our main empirical results. In particular, in Section 4, we consider several TD heuristics applied to toy graphs; and in Section 5, we consider TD heuristics applied to synthetic random graphs. Then, in Section 6, we describe the results of applying TDs to a carefully chosen suite of real-world social graphs. In Section 7, we prove a theoretical result connecting treewidth and treelength with the (very different) notion of δ-hyperbolicity, under an

assumption on the length of the longest geodesic cycle in the graph. Finally, in Section 8, we provide a brief discussion of results and conclusion.

2. BACKGROUND AND RELATED WORK

In this section, we will review relevant graph theory, TD ideas, and computational methods, as well as relevant related work.

2.1. Preliminaries on graph theory

Let $G = (V, E)$ be a *graph* with vertex set V and edge set $E \subseteq V \times V$. We often refer to graphs as *networks* and vertices as *nodes*, and we will model social and information networks by undirected graphs. We note that TDs are themselves graphs (constructed from other input graphs). The *degree* of a vertex v, denoted $d(v)$, is defined as the number of vertices that are adjacent to v (or the sum of the weights of adjacent edges, if the graph is weighted). The average degree is denoted $\bar{d}$. A graph is called *connected* if there exists a path between any two vertices. A graph is called a *tree* if it is connected and has no cycles. A vertex in a tree is called a *leaf* if it has degree 1. A graph $H = (S, F)$ is a subgraph of G if $S \subseteq V$, $F \subseteq E$. An *induced subgraph* of G on a set of vertices $S \subseteq V$ is the graph $G[S] := (S, S \times S \cap E)$. Unless otherwise specified, our analyses will always consider the *giant component*, i.e., the largest connected subgraph of G.

The *diameter* of a graph is the maximum distance between any two vertices, and the *eccentricity* of a vertex is the maximum distance between that vertex and all other vertices in the graph. Note that the maximum eccentricity of a graph is equal to the diameter. The *clustering coefficient* of a vertex is the ratio of the number of edges present among its neighbors to the maximum possible number of such edges; when we refer to the clustering coefficient of a network, we use the average of the clustering coefficient of all its vertices. A *cut* is a partitioning of a network's vertex set into two pieces. The *volume* of a cut is the sum over vertices in the smaller piece of the number of incident edges, and the *surface area* of a cut is the number of edges with one end-point in each piece. In this case, the *conductance* of a cut—one of the most important measures for assessing the quality of a cut—is the surface area divided by the volume (that is, we will be following the conventions used in previous work [77, 65]).

Finally, we will refer to the "core-periphery" structure of a network. Following prior work [77, 6, 65], we use the *k-core decomposition* to identify these core nodes. The *k*-core of a network G is the maximal induced subgraph $H \subseteq G$ such that every node in H has degree at least k. The *k*-core has the advantage of being easily computable in $O(V + E)$ time [16, 15, 17].

2.2. Preliminaries on tree decompositions

TDs are combinatorial objects that describe specialized mappings of cuts in a network to nodes of a tree. Although originally introduced in the context of structural graph theory (the proof of the graph minors theorem [87]), TDs have gained attention in the broader community due to their use in efficient algorithms for certain NP-hard problems. In particular, there are polynomial-time algorithms for solving many such problems on all graphs that have TDs whose width (defined below) is bounded from above by a constant [13, 20]. These algorithms have been applied to problems in constraint

satisfaction, computational biology, linear algebra, probabilistic networks, and machine learning [74, 75, 62, 107, 106, 78, 76, 68, 37].

Definition 2.1. A *tree decomposition (TD)* of a graph $G = (V, E)$ is a pair

$$(X = \{X_i : i \in I\}, T = (I, F)),$$

with each $X_i \subseteq V$, and T a tree with the following properties:

1. $\cup_{i \in I} X_i = V$,
2. For all $(v, w) \in E, \exists i \in I$ with $v, w \in X_i$, and
3. For all $v \in V, \{i \in I : v \in X_i\}$ forms a connected subtree of T.

The X_i are called the *bags* of the tree decomposition.

The third condition of the definition is a continuity requirement that allows the TD to be used in dynamic programming algorithms for many NP-hard problems.[1] It is equivalent to requiring that for all $i, j, k \in I$, if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.[2] The quality of a tree decomposition is often measured in terms of its largest bag size.

Definition 2.2. Let $\mathcal{T} = (\{X_i\}, T = (I, F))$ be a tree decomposition of a graph G. The *width* of $\mathcal{T}$ is defined to be $\max_{i \in I} |X_i| - 1$, and the *treewidth* of G, denoted $tw(G)$, is the minimum width over all valid tree decompositions of G. A tree decomposition whose width is equal to the treewidth is often referred to as *optimal*.

By this definition, trees have the minimum possible treewidth of 1 (their bags contain the edges of the original tree and thus have size 2); but, in contrast to δ-hyperbolicity (see Section 2.4), an n-vertex clique is the least treelike graph (attaining the maximum treewidth of $n - 1$). In fact, the only valid TDs of a clique have all vertices in a single bag. Because TDs remain valid under taking subgraphs (once you delete any vertices no longer present), if W is any complete subgraph of G, then every TD of G has some bag that contains all the vertices of W [32].

Two other canonical examples (to which we will return in detail) are the cycle and the grid, which have vastly differing treewidths. All cycles (regardless of the number of vertices) have treewidth 2 (see Figure 5 below). The $n \times n$ planar grid, however, has treewidth n, and thus, it is not treelike by this measure. Grids are particularly noteworthy

---

[1]Alternatively, the bags and edges of the TD form separators (cuts) in the graph. The set of vertices contained in any bag, or intersection of two adjacent bags, form a separator in G. This structural property is important because it allows TDs to be thought of as a method of organizing cuts in a network. This is also related to how the treewidth of a network is used to measure how treelike a network is. Intuitively, a tree has a treewidth of 1 because the graph can be separated by the removal of a single edge (or vertex) in the network, whereas a cycle requires two edges to be cut and thus has a treewidth of 2. TDs with large widths require larger numbers of vertices to separate a network into two disconnected pieces.

[2]A related aspect of the definition of a TD is the overlapping nature of the bags of a TD. Vertices in the graph will appear in many bags in the TD. This is particularly true of high-degree or high k-core nodes [6].

in the discussion of TDs due to a result (described in more detail below) showing that they are essentially the only obstruction to having bounded treewidth.[3]

Finding a TD for a given graph whose width is minimal (equal to the treewidth) is an NP-hard problem [93, 62]. Most methods (including those discussed here) for constructing TDs were designed to minimize width, because most prior work focused on using these structures to reduce computational cost for an algorithm/application.[4] Also, although treewidth is a graph invariant, TDs of a network are not unique, even under the condition of having minimum width. See, e.g., Figure 5, which shows several distinct minimum-width TDs of a cycle.

Finally, although it is not standard, we will abuse the term *width* to apply it directly to a bag of a tree decomposition (in which case, it takes the value of the cardinality of the set minus one), so that we can talk about the *maximum width* (which is the equivalent to the usual definition of width), and *median width* of a decomposition (which is the median of the widths of the bags). We will use the term *center* to refer to the bag (or bags) associated with the node(s) of minimum eccentricity in the tree underlying a TD, and the term *perimeter* for bags associated with nodes of relatively high eccentricity. We do this to help provide a framework for discussing the connection in many social and information networks between the core (resp. periphery) of the network and the central (resp. perimeter) bags of its TD computed with certain heuristics. Note that by definition, a tree will have at most two bags at its center.

2.3. Constructing tree decompositions

Here, we give a brief overview of existing algorithms for constructing TDs; more comprehensive surveys can be found in [62, 31]. The algorithms for finding low-width TDs are generally divided into two classes: "theoretical" and "computational." The former category includes, for example, the linear algorithm [26], which checks if a TD of width at most k exists (for a fixed constant k), and the approximation algorithms [11]. These are generally considered (practically) intractable due to very large hidden constants and complexity of implementation—e.g., Bodlaender's algorithm was shown [88] to have too high a computational cost even when $k = 4$. The approximation algorithms of Amir have been tested on graphs with up to several hundred vertices, but they require hours of running time even at this size scale. There has also been work on exact algorithms, the most computational of which is perhaps the QuickTree algorithm, which was tested on graphs with up to about 100 vertices and treewidth 11 [95]. Thus, in practice, most computational work requires the use of heuristic approaches (i.e., those that offer no worst-case guarantee on their maximum deviation from optimality). Because we are interested in applying TDs

---

[3]In particular, in the so-called grid minor theorem, Robertson and Seymour showed that every graph of treewidth at least k contains a $f(k) \times f(k)$ grid as a graph minor, for some integer-valued function f. The original estimate of the function f gave an exponential relationship between the treewidth and the grid size, and although several results greatly improved the relationship, the question of whether it held for any polynomial function f remained open for over 25 years. Recently, Chekuri and Chuzhoy proved that there is a universal constant $\delta > 0$ so that all graphs of treewidth at least k have a grid-minor of size $\Omega(k^\delta) \times \Omega(k^\delta)$ [36], resolving this conjecture.

[4]In the context of understanding the intermediate-scale structure of real networks and improving inference (e.g., link prediction, overlapping community detection, etc.), there are likely more appropriate objective functions, although their general identification and development is left as future work.

to real network data, we will focus on these "practical" algorithms in the remainder of this paper. We used INDDGO [57, 96], an open source software suite for computing TDs and numerous graph and TD parameters.

2.3.1. Chordal graph decomposition. A common method for constructing TDs is based on algorithms for decomposing *chordal graphs*.

Definition 2.3. A graph G is *chordal* if it has no induced cycles of length greater than three (equivalently, every cycle in G with length at least four, has a chord).

Chordal graphs are characterized by the existence of an ordering $\pi = (v_1, \ldots, v_n)$ of their vertices so that for each v_i, the set of its neighbors v_j with $j > i$ form a clique. This is a *perfect elimination ordering*, and it gives a straightforward construction for a TD (also called the clique-tree) of a chordal graph, with bags consisting of the sets of higher-indexed neighbors of each vertex.

For a general graph G, one common approach for finding TDs is to first find a chordal graph H containing G, then use the associated TD (as mentioned earlier, TDs remain valid for all subgraphs on the same vertex set). The typical approach is via *triangulation*, a process that uses a permutation of the vertex set (called the *elimination ordering*) to guide the addition of edges, which are referred to as *fill* edges. Chordal completions are not unique. For example, the complete graph formed on the vertices of G is chordal and contains G (although, it is a trivial or the "worst" triangulation, in the sense that it has the most fill edges and largest possible clique subgraph among all triangulations).

We will use the notation G_π^+ to denote the triangulation of G using ordering π. An outline of the process is given in Algorithm 1. The process for finding a TD T_π using an elimination order π and Gavril's algorithm ([53]) for decomposing chordal graphs is given in Algorithm 2. We may refer to the *width* of an ordering, by which we mean the treewidth of the chordal graph G_π^+. The literature includes several slight variants on Gavril's construction routine (such as Algorithm 2 in [31]), but the overall process and width of the TD produced is the same for each.

Algorithm 1 Triangulate a graph G into a chordal graph G_π^+

INPUT: Graph $G = (V, E)$, and $\pi = (v_1, \ldots, v_n)$, a permutation of V
OUTPUT: Chordal graph $G_\pi^+ \supseteq G$, for which π is a perfect elimination ordering
1: Initialize $G_\pi^+ = (V', E')$ with $V' = V$ and $E' = E$
2: **for** $i = 1$ to n **do**
3:     Let $N_i = \{v_j \mid j > i$ and $(v_i, v_j) \in E\}$
4:     **for** $\{x, y\} \subseteq N_i$ **do**
5:         **if** $x \neq y$ and $(x, y) \notin N_i$ **then**
6:             $E' = E' \cup \{(x, y)\}$
7:         **end if**
8:     **end for**
9: **end for**
10: **return** G_π^+

Perhaps surprisingly, there always exists some elimination ordering which produces an optimal TD (one of minimum width), and this may co-occur with high fill. The following theorem (see [31]) presents the connections between treewidth, triangulations, and elimination orderings.

Algorithm 2 Construct a TD T_π of a graph G using elimination ordering π and Gavril's algorithm

---

    INPUT: Graph $G = (V, E)$, π a permutation of V

    OUTPUT: TD $T_\pi = (X, (I, F))$ with (I, F) a tree, and bags $X = \{X_i\}$, $X_i \subseteq V$

1: Initialize $T = (X, (I, F))$ with $X = I = F = \emptyset$, $n = |V|$

2: Create an empty n-long array $t[]$

3: Use Algorithm 1 to create a triangulation G_π^+ using π.

4: Let $k = 1$, $I = \{1\}$, $X_1 = \{\pi_n\}$, $t[\pi_n] = 1$

5: **for** $i = n - 1$ to 1 **do**

6:     Find $B_i = \{$neighbors of π_i in $G_\pi^+\} \cap \{\pi_{i+1}, \ldots, \pi_n\}$

7:     Find $m = j$ such that $j \leq k$ for all $\pi_k \in B_i$

8:     **if** $B_i = X_{t[m]}$ **then**

9:         $X_{t[m]} = X_{t[m]} \cup \{\pi_i\}$; $t[\pi_i] = t[m]$

10:     **else**

11:         $k = k + 1$

12:         $I = I \cup \{k\}$; $X_k = B_i \cup \{\pi_i\}$

13:         $F = F \cup \{(k, t[m])\}$; $t[\pi_i] = k$

14:     **end if**

15: **end for**

16: **return** $T_\pi = (X, (I, F))$

Theorem 2.4. [31] *Let $G = (V, E)$ be a graph, and let $k \leq n$ be a nonnegative integer. Then, the following are equivalent.*

    1. G has treewidth at most k.

    2. G has a triangulation H s.t. any complete subgraph of H (clique) has at most $k + 1$ vertices.

    3. There is an elimination ordering π, such that G_π^+ does not contain any clique on $k + 2$ vertices as a subgraph.

    4. There is an elimination ordering π, such that no vertex $v \in V$ has more than k neighbors in G_π^+, which occur later in π.

Thus, if one can produce a "good" elimination ordering (i.e., one with a small maximum clique), it is easy to construct a low-width TD, and such an ordering always exists if the treewidth is bounded. The intuition behind fill-reducing orderings to minimize width follows from the idea that in order to produce a large clique that wasn't already in the network, one should have to add many fill edges.

    2.3.2. Ordering heuristics. Here, we describe the landscape of heuristics for creating elimination orderings, focusing on those used in our empirical evaluations. A more detailed analysis of heuristics as well as theoretical connections between chordal graphs and TDs is available [31]. The space of all possible elimination orderings is $O(n!)$ for a graph on n vertices, making it impractical to search using brute force techniques. One possibility for exploring the space is to apply a stochastic local search approach like simulated annealing, but because this is relatively slow, it is not common in practice.

    The first class of specialized methods is known as *triangulation recognition heuristics*, which include lexicographic breadth-first-search (LEX-M) and maximum cardinality search (MCS) [31, 90, 21, 22, 91, 98, 99]. These methods are guaranteed to provide a perfect elimination ordering for chordal graphs, so many believed they would produce low-fill

and/or low-width orderings for more general graphs. In [31], the authors report good results with respect to width when using these methods on graphs which are already chordal or have regular structures, but poor results compared to the greedy heuristics when even small amounts of randomness is added to the network. Further empirical evaluation [57] supports these claims. Additionally, these heuristics are too computationally expensive to run on very large graphs.

A large set of additional heuristics uses the idea of splitting the graph (using a small separator), recursively decomposing the resulting pieces, and then "gluing" the solutions into a single TD [11, 18, 29, 34, 75, 85]. "They are significantly more complex, significantly slower, and often give bounds that are higher than those of simple algorithms"[31]. We do, however, use a related approach that finds a set of nested graph partitions, but instead of decomposing the resulting pieces, it places the separators into an elimination ordering. This approach is called *nested dissection* [55, 56], and it is quite popular for computing fill-reducing orderings for sparse matrices in numerical linear algebra. The algorithm recursively finds a small vertex separator (bisector) in a graph, and it ensures that in the resulting elimination ordering, the vertices in the two components formed by the bisection all appear before the vertices in the separator. We use the "node nested dissection" algorithm implemented in METIS [69] (called through INDDGO), and we refer to this heuristic as METNND. In METIS, the recursion is stopped when the components are smaller than a certain size, and some version of minimum degree ordering is then applied to the remaining pieces. The software "grows" each bisection, using a greedy node-based strategy. Because the algorithm is searching for bisections, there is a tunable "balance" condition (determining how close to 50/50 the split needs to be), although for all computations reported in this study, we left the parameter at its default value.

Perhaps the most popular class of elimination ordering routines are *greedy* heuristics, named because they make greedy decisions to pick the subsequent node in the elimination ordering. There are innumerable variations, but the most common use two basic concepts: choosing a vertex to minimize *fill* (how many new edges will be added to the graph if a vertex is chosen to be next in the ordering); or choosing a vertex of minimum degree (low-degree vertices have small neighborhoods, which also limits the potential fill). When applied in their most rudimentary forms, these are the MINDEG [82] and MINFILL orderings. Both of these indirectly limit the size of cliques produced in the final triangulation (although they were originally designed to minimize the number of fill edges added during the triangulation, a quantity which is not always correlated). For additional heuristics combining these strategies and incorporating additional local information, see [31].

Even though keeping updated vertex degrees for MINDEG during triangulation (greedy orderings make their decisions based on a partially triangulated graph at each step) is significantly less computationally intensive than computing current vertex fills for MINFILL, there have been efforts to reduce further the complexity. In particular, the approximate minimum degree or AMD heuristic [9, 8]. This heuristic computes an upper bound on the degree of each node in each pass using techniques based on the quotient graph for matrix factorization, and it has been shown to be significantly faster and of similar quality (in terms of fill and width minimization). We use AMD interchangeably with the traditional MINDEG, especially on larger networks.

2.4. Additional related work

For completeness, we provide here a brief overview of the large body of additional related work. As already mentioned, TDs played an important role in the proving of the graph

minor theorem [87], but they have also become popular in theoretical computer science, as many NP-hard optimization problems have a polynomial time algorithm for graphs with bounded treewidth [13]. In addition, bounding the treewidth of the underlying graph of probabilistic graphical models allows for fast inference computations [71]. Additional overviews of TDs and their uses in discrete optimization are available [25, 27, 28, 30]; and one can also learn more about the uses of these methods in numerical linear algebra and sparse matrix computations [24], as well as connections with triangulation methods: triangulation of minimum treewidth [10], empirical work on treewidth computations [73], the minimum degree heuristic and connections with triangulation [23], and a survey of triangulation methods [61]. Finally, the treewidth of random graphs for various parameter settings has been studied [103, 52].

A different notion of treelikeness is provided by δ-hyperbolicity.[5] Early, more mathematical work did not consider graphs and networks [59, 7], but more recent more applied work has [67, 66, 38]. Computing δ exactly is very expensive [5], and sampling-based methods to approximate it provide only a very rough guide to its value and properties [6]. For many references on δ-hyperbolicity in network analysis, see [38] (and the more recent work [102]) and references therein. There has been work on trying to relate hyperbolicity and TD-based ideas, often going beyond treewidth to consider other metrics such as treelength or chordality or the expansion properties of the graph [35, 105, 47, 79, 58, 72, 48, 2, 1].

Recent work in network analysis and community structure analysis has pointed to some sort of "core-periphery" structure in many real networks [77, 94, 89, 6, 65]; and recently this has been related to the k-core decomposition—see, e.g., [6] and references therein. The k-core decomposition is of interest more generally, and additional references for k-core decompositions, including their use in visualization and in larger-scale applications, include [92, 63, 64, 60, 14, 39, 42]. Questions of well-connected or expander-like cores are of particular interest in applications having to do with diffusion processes, influential spreaders, and related questions of social contagion [70, 101].

There are a few other works that have used TDs to investigate the structural properties of social and information networks: e.g., to look at the treelikeness of Internet latency and bandwidth [84]; to compare hyperbolicity and treewidth on internet networks [45]; and to examine the relationship between hyperbolicity, treewidth, and the core-periphery structure in a much wider range of social and information networks [6]. In particular, [45] concludes that the hyperbolicity is small in the networks they examined but the treewidth is relatively large, presumably due to a highly connected core; and [6] concludes that many real social and information networks do have a treelike structure, with respect to both metric-based hyperbolicity and (in spite of the large treewidth) the cut-based TDs, that corresponds to the core-periphery structure of the network. Finally, very recently we became aware of [80] and [43].

3. NETWORK DATASETS

We have examined the empirical performance of existing TD heuristics on a broad set of real-world social and information networks as well as a large corpus of synthetic

---

[5]Our prior work focused on the use of δ-hyperbolicity [5, 6, 4]. It can be a useful tool for describing and analyzing real networks, even though it is expensive to compute, but aside from our theoretical result in Section 7 relating it to treewidth and treelength, it is not our focus in this work.

Network	n_c	k_l	k_m	$\bar{d}$	$\bar{C}$	D
ER Random Graphs						
ER (1.6)	3210	1	2	2.16	0.00	38
ER (1.8)	3617	1	2	2.28	9.30×10^{-4}	34
ER (2)	4001	1	2	2.39	9.11×10^{-4}	30
ER (4)	4879	1	3	4.05	8.96×10^{-3}	15
ER (8)	4998	1	5	8.04	1.59×10^{-3}	7
ER (16)	5000	4	11	16.1	3.13×10^{-3}	5
ER (32)	5000	7	23	32.1	6.39×10^{-3}	4
PL Random Graphs						
PL (2.50)	4895	1	4	2.78	2.46×10^{-3}	18
PL (2.75)	4650	1	2	2.43	6.99×10^{-4}	22
PL (3.00)	4071	1	2	2.24	1.18×10^{-3}	29
SNAP Social Graphs						
CA-GRQC	4158	1	43	6.46	.665	17
CA-ASTROPH	17903	1	56	22.0	.669	14
AS 20000102	6474	1	12	3.88	.399	9
Gnutella 09	8104	1	10	6.42	.0137	10
EMAIL-ENRON	33696	1	43	10.7	.708	13
FB Social Graphs						
FB-CALTECH	762	1	35	43.7	.426	6
FB-HAVERFORD	1446	1	63	82.4	.327	6
FB-LEHIGH	5073	1	62	78.2	.270	6
FB-RICE	4083	1	72	90.5	.300	6
FB-STANFORD	11586	1	91	98.1	.252	9
Miscellaneous Graphs						
POWERGRID	4941	1	5	2.67	0.107	46
POLBLOGS	1222	1	36	27.4	0.360	8
PLANARGRID	2500	2	2	3.92	0.00	98
ROAD-TX	1379917	1	3	1.39	0.0209	1054
WEB-STANFORD	281903	1	71	8.20	2.89×10^{-3}	674

Table I Statistics of analyzed networks: nodes in giant component n_c, k_l the lowest k-core, k_m the maximum k-core, average degree $\bar{d} = 2E/N$, average clustering coefficient $\bar{C}$, and diameter D.

graphs. The real-world networks have been chosen to be representative of a broad range of networks, as analyzed in prior work [77, 6, 65], and the synthetic graphs have been chosen to illustrate the behavior of TD methods in controlled settings. See Table I for a summary of the networks we have considered. The real-world graphs are connected, but we are interested in parameter values for the synthetic graphs which might cause the instances to be *disconnected*. In these cases, we work with the giant component, and the statistics in Table I are for this connected subgraph.

Erdős–Rényi (ER) graphs. Although ER graphs are often criticized for their inability to model pertinent properties of realistic networks, *extremely* sparse ER graphs have several structural inhomogeneities that are important for understanding treelike structure in realistic

networks [6].[6] In particular, in the extremely sparse regime of $1/n < p < \log(n)/n$, ER graphs are (w.h.p.) not even fully connected; ER graphs in this regime have an upward-trending network community profile (NCP) [77]; with respect to their k-core structure, a shallow (but nontrivial) core-periphery structure emerges [6, 83]; and with respect to their metric properties (as measured with δ-hyperbolicity), graphs in this regime have nontrivial treelike properties [6]. Following previous work [6], we set the target number of vertices to $n = 5,000$, and we choose $p = \frac{d}{n}$ for various values of d from $d = 1.6$ to $d = 32$. We denote these networks using ER(d). Table I clearly shows that, as a function of *increasing* d, i.e., increasing p, the size of the giant component increases to 5,000, the number of edges increases dramatically, the clustering coefficient remains close to zero, the average degree $\bar{d}$ increases, and the diameter decreases dramatically.

Power Law (PL) graphs. We also considered the model in [41], an ER-like random graph model parameterized to have a power law degree distribution (in expectation) with power law (or heterogeneity) parameter γ, which we vary between 2 and 3. We denote these networks using PL(γ). We consider values of the degree heterogeneity parameter $\gamma \in \{2.50, 2.75, 3.00\}$. Table I shows that, as a function of *decreasing* γ, the size of the giant component increases, the average degree $\bar{d}$ increases, and the diameter decreases. Although not shown in Table I, as γ decreases, PL graphs also form a rather prominent, and moderately deep, k-core structure [6]. These are all trends that parallel the behavior of ER as d increases.[7]

SNAP graphs. We selected various social/information networks that were used in the large-scale empirical analysis that first established the upward-sloping NCP and associated nested core-periphery for a broad range of realistic social and information graphs [77]. These are available at the SNAP website [97]. In particular, the networks we considered are CA-GRQC and CA-ASTROPH (two collaboration networks), AS20000102 (an autonomous system snapshot), GNUTELLA09 (a peer-to-peer network from Gnutella), EMAIL-ENRON (an email network from the Enron database), as well as the Stanford Web network WEB-STANFORD and the Texas road network ROAD-TX. These networks are very sparse, e.g., fewer than ca. 10 edges per node; and they exhibit a substantial degree of heterogeneity, moderately high clustering coefficients (except for GNUTELLA09, WEB-STANFORD, and ROAD-TX), and moderately small diameters. In addition, although not presented in Table I (and with the exception of ROAD-TX), these graphs have a much stronger core-periphery structure, as measured by the k-core decomposition, than typical synthetic networks [6, 65, 77].

Facebook networks. We selected several representative Facebook graphs out of ca. 100 Facebook graphs from various American universities collected in ca. 2005 [100]. These datasets range in size from around 700 vertices (FB-CALTECH) to approximately 30,000 vertices (FB-TEXAS84). In particular, we examine FB-CALTECH, FB-RICE, FB-HAVERFORD, FB-LEHIGH, and FB-STANFORD in this article. These networks all arise via similar generative procedures, and thus there are strong similarities between them. There are a few distinctive networks, however, that are worth mentioning. In particular, several universities

---

[6]The same is true for many other less unrealistic random graph models, assuming their parameters are set to analogously sparse values (which they are often not).

[7]For sparse ER graphs, this happens when there are not enough edges for concentration of measure to occur, i.e., for empirical quantities such as the empirical degrees to be very close to their expected value. For PL graphs, an analogous lack of measure concentration occurs due to the exogenously specified heterogeneity parameter γ.

(FB-Caltech, FB-Rice, FB-UCSC) have a particularly strong resident housing system, and it is known that this manifests itself in structural properties of the graphs [100]. Following, we will use the metainformation associated with this housing system to provide ground-truth clusters/communities for comparison and evaluation.[8] One important characteristic to observe from Table I is that these Facebook networks, although sparse, are much denser than any of the SNAP graphs we consider or that were considered previously [77].[9]

Miscellaneous networks. We also selected a variety of real-world networks that, based on prior work [77, 6, 65], are known to have very different properties than the SNAP social graphs or the Facebook social graphs. In particular, we consider Polblogs, a political bloggers network [3] (a graph constructed from political blogs, which are linked), the Western US power grid PowerGrid [104]; and a two-dimensional 50×50 planar grid PlanarGrid.

4. TREE DECOMPOSITIONS OF TOY NETWORKS

In this section, we will describe the results of using a variety of TD heuristics on a set of very simple "toy" networks, on which the optimal-width TDs are known. The five toy networks we consider are a binary tree (SmallBinary), a small section of the two-dimensional planar grid (SmallPlanar), a cycle (SmallCycle), a clique (SmallClique), and an Erdős–Rényi graph with an edge probability of $p = 0.5$ (SmallER). Each of these networks has 100 nodes (except for SmallCycle, which has only 10 nodes—the reason for this is that the principal change by having a larger cycle is that the eccentricity of the decompositions becomes much larger, which simply makes it more difficult to visualize—and SmallBinary, which has 128 nodes to maintain symmetry). In Figure 1, we provide visualizations[10] of each of the five networks.

These very simple network topologies illustrate in a controlled way the behavior of different TD heuristics in a range of settings. For example, while SmallBinary is a tree, the other graphs are not; the two-dimensional grid is quite different from a tree, as is SmallCycle (although, from the treewidth perspective it is fairly close to a tree), and both have high-quality well-balanced partitions; and both SmallClique and SmallER are expanders (not constant degree expanders, but expanders in the sense that they don't have any good partitions) and thus very non-treelike (from the TD perspective), but each has important differences with respect to their respective TDs. We will focus on which types of structures different heuristics tend to capture, as well as how different heuristics deal with nodes (not bags) which are associated with the core or periphery of the original network. Importantly, these toy networks have basic constructions, and they (mostly) have known optimal width TDs—e.g., SmallPlanar and SmallCycle have several known equivalent minimum width TDs—and the SmallER network serves to illustrate some of the effects of randomness on a TD. The insights we obtain here can be used to interpret the output of TD heuristics in much more complex synthetic and real networks.

---

[8]See [65] for how this affects the NCP of these networks.

[9]Among the differences caused by the much higher density of Facebook networks is that these networks have a much deeper k-core structure than the other real networks, and they tend to lack deep cuts, e.g., they lack even good very imbalanced partitions such as those responsible for the upward-sloping NCP [77, 65].

[10]These and other visualizations were created with the GraphViz command *neato* [51], with the help of [44, 81].
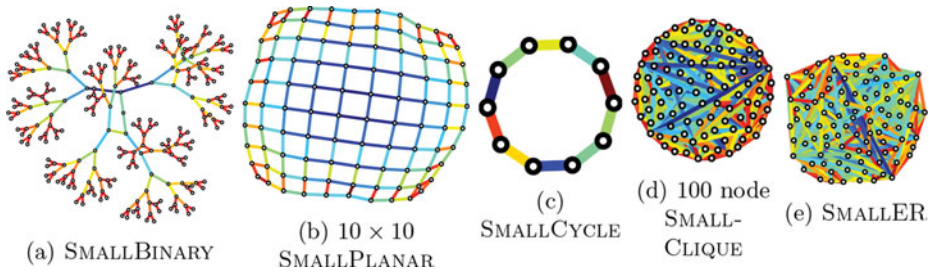
Figure 1 A set of small networks. Edges are colored by their length in the planar embedding.

4.1. TD properties of toy networks

In Figures 2, 3, and 4, we show visualizations of TDs produced by various heuristics (the greedy MINDEG in Figure 2; the METNND, nested node dissection via METIS, in Figure 3; and LEXM in Figure 4) for each of these five toy networks. In these visualizations, the size of the bag corresponds with the bag's width, and the coloring is based on the fraction of edges present in the induced subgraph of the bag. In particular, if the nodes in the bag form a clique in the original network, then the fraction of edges present is 1.0 and the bag is dark red; whereas if the the nodes are completely disconnected in the original network, the bag is dark blue.

From these figures, we see substantial differences between the TDs that different heuristics generate for these five toy networks. All heuristics give the same uninteresting results for SMALLCLIQUE; but for all of the other networks, including SMALLBINARY, there are differences in the decompositions produced by the different heuristics. Consider SMALLPLANAR, SMALLCYCLE, and SMALLER. For both SMALLPLANAR and SMALLER, MINDEG and METNND return TDs with several prominent branches, whereas LEXM returns a path for the TD. For MINDEG, this is due to the tendency of the algorithm to pick low-degree nodes on the "outside" of the network and then work its way around the outside of the network. For METNND, this is due to the tendency of the algorithm to cut the networks repeatedly into smaller pieces and then recursively "eat away" at these smaller pieces to form the TD. On the other hand, the LEXM heuristic works to produce a minimal triangulation using lexicographic labelings along paths. This often results in a pathlike TD, as the algorithm uses a breadth-first search through the network. For SMALLCYCLE, METNND returns a "branchy" TD, whereas both MINDEG and LEXM return pathlike TDs.
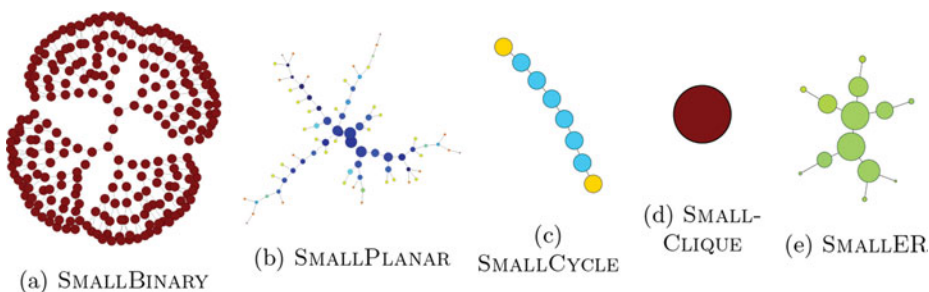
Figure 2 Greedy MINDEG TDs of toy networks. Bags are colored by the fraction of possible edges present in the bag, with red being denser and blue being less dense.

(a) SMALLBINARY   (b) SMALLPLANAR   (c) SMALLCYCLE   (d) SMALL-CLIQUE   (e) SMALLER
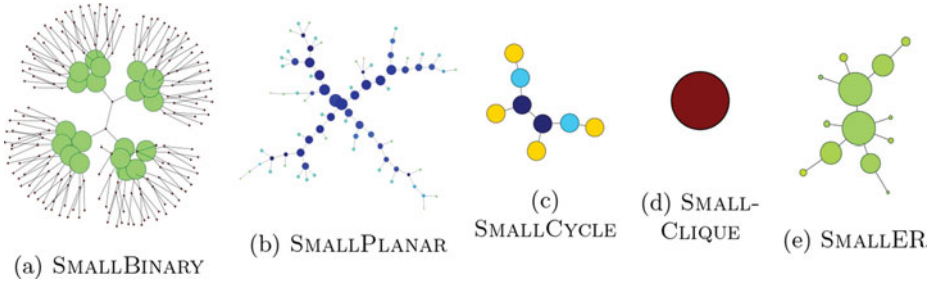
Figure 3 METNND (nested node dissection via METIS) TDs of toy networks. Bags are colored by the fraction of possible edges present in the bag, with red being denser and blue being less dense.

(a) SMALLBINARY   (b) SMALLPLANAR   (c) SMALLCYCLE   (d) SMALL-CLIQUE   (e) SMALLER
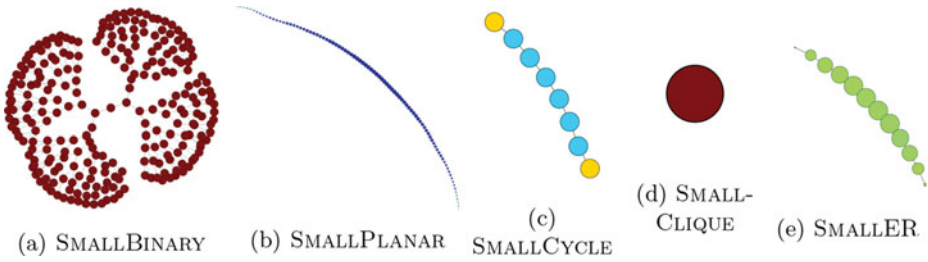
Figure 4 LEXM TDs of toy networks. Bags are colored by the fraction of possible edges present in the bag, with red being denser and blue being less dense.

More quantitatively, in Tables II and III, we provide basic statistics for TD heuristics applied to each of these networks. Table II shows a summary of our results for the the (maximum) width of TDs produced by various heuristics (for SMALLER, the width given is averaged over five different instantiations of the network). We ignore the issue of tie-breaker choices (e.g., in MINDEG, choosing among nonunique minimum degree nodes). On the whole, the heuristics do a good job of finding optimal width TDs on SMALLBINARY, SMALLCLIQUE, and SMALLCYCLE (with the exception of METNND). The greedy heuristics have trouble finding the optimal width TD on SMALLPLANAR, whereas LEXM and MCS both find an optimal decomposition on the grid. On SMALLER, we observe that the greedy heuristics and METNND outperform LEXM and MCS; this is in agreement with previously reported results [31]. Table III shows a summary of our results for the *median* width of TDs produced by various heuristics (as defined in Section 2.2). The median width is potentially more useful for revealing structure in realistic network data since, e.g., it can be used to see whether a TD is dominated by larger bags or by smaller bags.[11] If a network is dominated by bags of small size (such as SMALLBINARY, SMALLCYCLE, SMALLPLANAR),

[11]Using medians rather than eccentricity can result in different central bags. However, in most of the networks that we studied, the results were very similar. In particular, the biggest changes occurred in the FB networks where the median shifted towards the heavier end of the pathlike TD. However, these bags were still a part of the thick trunk of the network, and thus, the results were very similar. In other networks, the median bag was very close the central eccentric bag, and the main difference is that the median bag tended to have more whisker branches (a branch consisting of one or two bags of small width). This does not substantially change any of our analysis.

Network	n	W_{MINDEG}	W_{MINFILL}	W_{NND}	W_{MCS}	W_{LEXM}
SmallBinary	128	1	1	3	1	1
SmallPlanar	100	13	13	14	10	10
SmallCycle	10	3	3	3	3	3
SmallClique	100	100	100	100	100	100
SmallER	100	86	85	86	91	89

Table II TD heuristic *maximum* widths. The widths of the SmallClique and SmallER are relatively large (they grow linearly with the network size), the width of SmallPlanar network is of an intermediate size (they grow with the square root of the network size), and the widths of SmallBinary and SmallCycle are small (they stay constant with the size of the networks). The greedy heuristics find smaller width decompositions on SmallER, while lexm and mcs perform better on SmallPlanar.
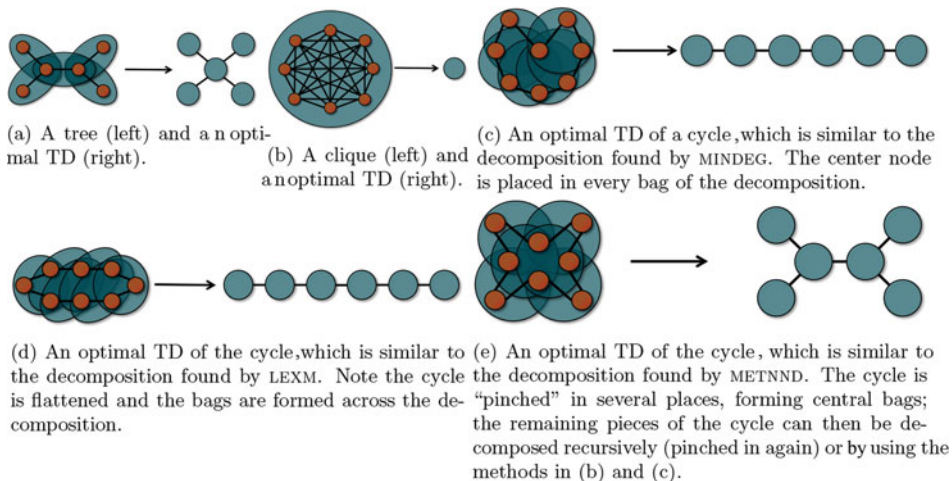
Network	n	$\tilde{W}_{\text{MINDEG}}$	$\tilde{W}_{\text{MINFILL}}$	$\tilde{W}_{\text{NND}}$	$\tilde{W}_{\text{MCS}}$	$\tilde{W}_{\text{LEXM}}$
SmallBinary	128	1	1	1	1	1
SmallPlanar	100	5	5	5	10	8
SmallCycle	10	3	3	3	3	3
SmallClique	100	100	100	100	100	100
SmallER	100	52	51	49	85	80

Table III TD heuristic *median* widths. This quantity is much smaller than the corresponding widths in several of the networks (although it remains large with SmallER), indicating that these networks are dominated by bags that are much smaller than the largest bag in the network.

depending on the internal structure of the bag, this can indicate several things. For example, the small bags could consist of tight clusters or cliques, indicating that the network has many tightly connected but *small* groups of nodes. Alternatively, if a small bag's structure is mostly disconnected, this may indicate the bag is related to small cycles (an example is given below).

For SmallCycle and SmallPlanar, the small bags are cyclical, whereas for SmallBinary, the small bags all consist of 2-cliques. SmallClique and the SmallER have large median widths (though this is trivial in the case of the clique). The 100-clique is both trivial and too large of a clique to be realistic, but SmallER has interesting bags. The results in Table II show the small median widths of SmallPlanar, SmallCycle and SmallBinary and the large median widths of SmallER and SmallClique. Table II also demonstrates that, although there are differences in the widths of the TDs produced by the heuristics, these differences are reasonably small.[12]

[12]We will see that most real networks have small median width, with smallest bags dominated by cliques, intermediate bags dominated by cycles, and with large, connected, central bags resemble bags of SmallER.

(a) A tree (left) and an optimal TD (right).

(b) A clique (left) and an optimal TD (right).

(c) An optimal TD of a cycle, which is similar to the decomposition found by MINDEG. The center node is placed in every bag of the decomposition.

(d) An optimal TD of the cycle, which is similar to the decomposition found by LEXM. Note the cycle is flattened and the bags are formed across the decomposition.

(e) An optimal TD of the cycle, which is similar to the decomposition found by METNND. The cycle is "pinched" in several places, forming central bags; the remaining pieces of the cycle can then be decomposed recursively (pinched in again) or by using the methods in (b) and (c).
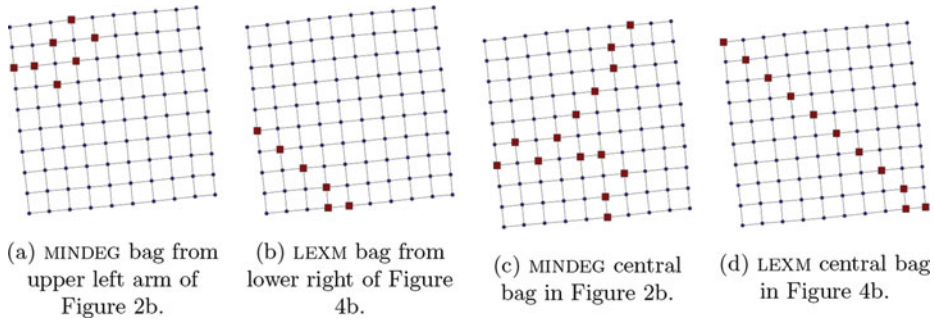
Figure 5 Example TDs. The tree and the clique have a standard optimal TD. The cycle has many possible minimum width TDs, though all place disconnected nodes in the bags.

4.2. TDs on cliquelike and cyclelike toy networks

An important aspect of TD heuristics is the difference between their behavior on (denser) cliquelike graphs and (sparser) cyclelike graphs. In Figure 5, we illustrate this. First, for reference, in Figures 5a and 5b we give canonical minimum width TDs for a tree and a clique. To understand the difference between cycles and cliques, recall that there are many ways of producing a TD on a cycle (three of these are illustrated in Figures 5c, 5d, and 5e). One simple way is to produce a tree that is a path. This can be done by taking a node v and placing it and its two neighbors in a bag at one end of the path. Then, keeping v in every bag, progress around the cycle sequentially forming the next bag by including v and the two nodes of the next edge (see Figure 5c). Another method produces a path by "flattening" the cycle, and places each edge in a bag with the node from the other side (see Figure 5d). The METNND heuristic "pinches" the cycle at a few points, and the produces branches from each of those points (see Figure 5e).

There are many differences between these TD heuristics, but an important point is that the nodes in the cycle must be placed in bags with nodes they are *not* neighbors of the original graph. Different TD heuristics are very different in terms of how they make this decision, and its effect can be seen in the TDs constructed by these heuristics.

Another important consideration is the interior structure of each bag that is produced by a TD heuristic. Recall that in SMALLCLIQUE, the only valid TD (which does not contain unnecessary bags) is a single bag containing the entire network. Relatedly, if the network is a *k-tree*, formed by overlapping cliques (rather than overlapping edges, as in a normal or 2-tree), then the TD will have bags that consist of the individual cliques. Thus, with cliques, it is the *local structure* (local in the original graph, in the sense that it is driven by neighbors of a given node in the original graph, in contrast with what is going on in, e.g., SMALLCYCLE) that drives the bag formation. With cycles, on the other hand, this local structure is partially "lost" in the bags of the TD. This is of interest because, as already mentioned, the interior structure of bags of different widths is important for understanding what is creating the properties of the TD.

(a) MINDEG bag from upper left arm of Figure 2b.

(b) LEXM bag from lower right of Figure 4b.

(c) MINDEG central bag in Figure 2b.

(d) LEXM central bag in Figure 4b.

Figure 6 Representative bags from an arm in MINDEG and an arm in LEXM, as well as the associated central bags. In MINDEG, each arm progresses from a different corner of SMALLPLANAR. However, when these bag lines converge, the central bags end up containing pieces of each line, as in Figure 6c. In LEXM, the line proceeds diagonally across the grid from the lower-left corner to the upper right in a regular manner, as in Figure 6d. This results in smaller central bags and produces a path decomposition. See the main text for more details.

As an example, we observe that, for all of the heuristics, the larger bags on SMALLPLANAR have many disconnected nodes and only a few edges. This is a signature of "cyclical" behavior; and, indeed, from the TD perspective, the grid "looks like" a set of small, regular, overlapping cycles. The structure of the TD is formed by the heuristic's method of moving across the grid and closing cycles. This suggests that a simple metric to measure whether the interior of a bag is driven by cycles or by small, tightly connected clusters: measure the fraction of edges present in the bag, i.e., the *edge density* of the bag. (We will do this following, and this is why we color many of the visualizations by the density of the bag.)

4.3. TDs on well-partitionable and poorly partitionable toy networks

SMALLPLANAR (for which there exist good well-balanced partitions) and SMALLER (for which there do not exist good well-balanced partitions) also illustrate differences between the TD heuristics. For both graphs, the greedy heuristics and METNND have significantly smaller median widths than maximum widths. This is indicative of heterogeneity in the network: there are nodes that are so entangled with other nodes that they must appear together in a large bag, but there are also nodes that are connected to only a small number of other nodes and need to appear in only a few very small bags. This can partially be explained by the tendency of the greedy heuristics to work from the "boundary" of the graph (e.g., boundary nodes have smaller degrees) and to pick points to "eat into" the graph.

This is illustrated in Figure 6 for SMALLPLANAR. Using MINDEG as an example, recall that heuristic works by successively picking a minimum degree node in the network; thus, when applied to SMALLPLANAR, it will pick each of the corner vertices of the grid first. This then forms small bags at each corner and, depending on whether it is picking nonunique nodes at random or in an ordered fashion, it will then proceed to work in from the periphery of the network. Indeed, in Figure 2 and 3, we see that the TDs for these heuristics have four major arms with small leaves containing nodes from the border of the grid. Figure 6 provides a visualization of where the nodes from these bags (one of the peripheral bags and one of the central bags in the TD) are in SMALLPLANAR. (See, in particular Figures 6a and 6c for MINDEG.)

The LEXM and MCS heuristics, in contrast, find the minimum possible width for the grid, but the TDs—as illustrated in Figure 4 for LEXM—that are produced are long, pathlike trunks. This is due to they way that LEXM picks a starting node and then works across the graph in the style of a breadth-first search. With SMALLPLANAR, it starts at one corner and then moves across the network to form a minimal triangulation. Although the (maximum) width is minimal, the median widths of these networks are relatively large, because most of the bags are roughly the same size (see Figures 6b and 6d for the results of LEXM).

With SMALLER (which is harder to visualize because it doesn't embed well in two dimensions), the MINDEG and METNND algorithms also eat in from the boundary of the network, where, here, "boundary" means nodes with *slightly* smaller degrees or *slightly* better cuts (slightly smaller due to random fluctuations). As with the very different SMALLPLANAR, this produces several arms and then a few central bags. In SMALLER, the greedy heuristics produce a better TD in terms of width than the LEXM and MCS heuristics, both search based heuristics. These similarities and (substantial) differences between TD heuristics in SMALLER (compared with SMALLPLANAR) are apparent in Figures 2, 3, and 4.

4.4. Summary of TD results on toy networks

Overall, the greedy heuristics, e.g., MINDEG or METNND, seem to produce a better representation of the large-scale structure of SMALLPLANAR and SMALLER than the LEXM and MCS heuristics in two ways. In SMALLER, the greedy heuristics find decompositions with both smaller maximum as well as smaller median widths. (Because most real networks have a randomized aspect to their generation, this indicates that greedy heuristics might be more useful.) On SMALLPLANAR, the median width is smaller and the greedy heuristics do a better job of "capturing" all four corners of the grid. In other words, the resulting tree decomposition has four branches, each of which is tied to a specific corner of the network, while LEXM and MCS TDs capture two of the corner structures. (Although the maximum width is smaller with LEXM and MCS, the ability to capture what is an obvious visual feature of a simple network is of potential interest.) In the rest of the article, we will be considering significantly larger and more complicated networks than these toy examples. With these larger networks, the METNND and AMD heuristics, as implemented using INDDGO [57], are the most scalable, compared with the basic greedy algorithms (MINDEG or MINFILL). The AMD heuristic is very related to the MINDEG heuristic (recall that AMD picks minimum nodes based on an easy-to-compute *approximation* of node degree), and it gives similar results to MINDEG. The the most consistent difference between the two heuristics seems to be the number of central/overlapping bags produced. Thus, we will often show results only for the AMD heuristic as a matter of visual convenience.

5. TREE DECOMPOSITIONS OF SYNTHETIC NETWORKS

In this section, we will describe the results of using a variety of TD heuristics on a set of synthetic networks. We focus our attention on two simple classes of random graphs: the popular ER random graphs (in Section 5.1); and a power law (PL) extension of the basic ER model (in Section 5.2). (We emphasize, though, that similar qualitative results also hold for many other random graph models—in their extremely sparse regimes.) This will allow us to begin to understand how TDs behave in random graph models with a very simple random structure. Importantly, we will focus on *extremely* sparse graphs. For the ER model, this means values of the connection probability p that lead to the graph not even

being fully connected (in which case we will consider the giant component), whereas for the PL model, this means values of the degree heterogeneity parameter γ that are typically used to describe many realistic networks and that lead to analogously sparse graphs.
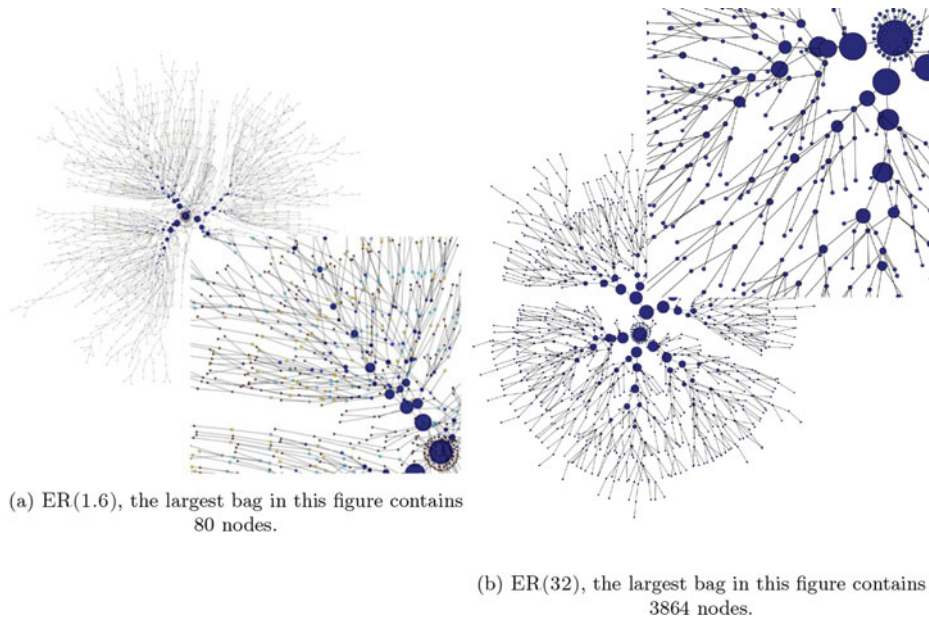
ER graphs are often presented as "strawmen," because they obviously do not provide a realistic model for many aspects of real-world networks (e.g., the heavy-tailed degree distributions and the nonzero clustering coefficient present in many real networks). Indeed, "vanilla ER" graphs that are often considered (e.g., ER graphs with densities that are sufficiently large that the graph is fully-connected) are *not* treelike—neither by the metric notion of δ-hyperbolicity nor by the cut-based notion of TDs. Recent work has shown, however, that with respect to their large-scale structure, *extremely* sparse ER networks do capture several subtle but ubiquitous properties of interest in realistic networks: first, the small-scale versus large-scale isoperimetric structure of the NCP [77, 65]; second, a size-resolved version of δ-hyperbolicity that is consistent with large-scale metric treelikeness [6]; and third, a nontrivial core-periphery structure with respect to k-core decompositions [6]. (In particular, in the sparsest regime of the ER networks that we consider, ER(1.6), a very shallow core-periphery structure appears—whereas none exists at the higher densities.) Importantly, for all three of these properties, similar results were seen with other random graph models, such as PL random graphs in the regime of the degree heterogeneity parameter that is commonly used. Prior work has also provided evidence that these extremely sparse random graphs have nontrivial treelike structure (at least relative to much denser ER graphs) when viewed with respect to the cut-like notion of treelikeness [6].

Here, we provide a much more detailed analysis of this phenomenon for TD heuristics applied to ER and PL graphs. We will be particularly interested in similarities between extremely sparse ER graphs and PL graphs with respect to the core-periphery structure (e.g., from k-core and related decompositions) of a network. Among other things, we show that this core-periphery structure is captured with the AMD TD. Of particular interest is the how the core-periphery structure relates to central (low eccentricity) or perimeter (high eccentricity) bags in the TD.

5.1. TDs of ER networks

Here, we give a summary of the empirical results of an analysis of TDs on ER random graphs, with an emphasis on the behavior as the connectivity parameter p is varied. In the very sparse to extremely sparse regime, ER networks have nontrivial global structural changes as p is varied [50, 33]. In particular, for our subsequent results, there are three regimes of p that are of interest: if $p < \frac{1}{n}$, then the largest connected component is $O(\log n)$ in size, and the small components are likely trees; if $\frac{1}{n} < p < \frac{\log n}{n}$, then the graph has a giant component (i.e., a constant fraction of the size of the network is connected), and the remaining small components of size $O(\log n)$ are likely trees; and if $p > \frac{\log n}{n}$, then almost surely the network is fully connected, the degrees are very near their expected value, and there are no good-conductance clusters (of any size). We are interested in these last two regimes, and we consider synthetic graphs (ER(1.6) through ER(32)—values of p between 1.6/5, 000 and 32/5, 000, for graphs with $n = 5, 000$ nodes) that go from extremely sparse to somewhat denser. Table I provides basic statistics for these graphs.

5.1.1. Visualization and basic statistics. We start with Figure 7 and Table 4a, which show the basic features of the TDs of ER networks. Figure 7 presents a visualization of part of the output of a TD with the AMD heuristic, colored by density of bag subgraph, for

(a) ER(1.6), the largest bag in this figure contains 80 nodes.

(b) ER(32), the largest bag in this figure contains 3864 nodes.

Figure 7 Visualization of ER(1.6) and ER(32) AMD tree decompositions, colored by the density of the bag subgraph. For visualization purposes, the two networks are not drawn to the same scale. The bags in ER(32) have widths that are approximately 50 times larger that that of ER(1.6). The blowups show the upper-left corner of the visualization in greater details. The blowups show the color of some of the smaller bags that are in the peripheral part of each TD. In ER(1.6), many of the very small bags are red (meaning they contain a clique, the vast majority of which are simply a single edge). Slightly larger bags are light blue or yellow (indicating an edge density of ca. 0.25 (light blue) to ca. 0.75 (yellow)). In ER(32), all of the bags of the TD are dark blue, indicating that these bags are all very sparse, regardless of whether they are peripheral or central to the TD. The statistics in Table 4a confirm this.

the sparsest (ER(1.6)) and densest (ER(32)) networks in our ER suite. Results are similar to those of METNND. Observe that there is a much greater heterogeneity in the density of bags for ER(1.6) than for ER(32). For the former, there are many small bags that are cliques; whereas for the latter, there are fewer small bags, and the bags are much sparser in general. This suggests (and we have verified by inspection) that the sparser ER(1.6) has greater

Network	N_{AMD}	E_{AMD}	W	$\tilde{W}$	$\tilde{D}$	Network	N_{AMD}	E_{AMD}	W	$\tilde{W}$	$\tilde{D}$
ER(1.6)	3127	44	79	1	1.0	PL(2.5)	4672	32	219	1	1.0
ER(1.8)	3457	38	157	1	1.0	PL(2.75)	4500	39	148	1	1.0
ER(2)	3760	38	235	2	0.67	PL(3.0)	3974	36	96	1	1.0
ER(4)	3777	35	1093	3	0.40						
ER(8)	2787	29	2208	8	0.20						
ER(16)	1856	28	3142	17	0.12						
ER(32)	1136	22	3863	33	0.06						
(a) ER networks						(b) PL networks					

Table IV Basic AMD TD statistics for ER and PL networks: N_{AMD} gives the number of bags in the TD, E_{AMD} gives the maximum eccentricity (diameter) of the TD, W and $\tilde{W}$ are the maximum and median width of the TD, and $\tilde{D}$ is the median bag density.
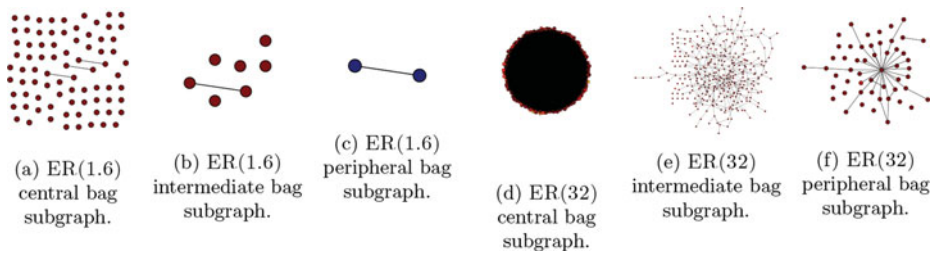
structural heterogeneity than the denser ER(32). A more detailed understanding of this can be obtained from the summary statistics in Table IVa. Several observations are worth making. First, the number of bags in the TD tends to decrease as the density p increases (with the exception of the sparsest regime, where the giant component is smaller). This is because the network is mostly placed into one bag, and only a few bags are needed to take care of the remaining nodes. Second, the TD itself, viewed as a graph, has smaller diameter as the density p increases. Third, the maximum and median width increases with the average degree of the network. Indeed, the width increases quickly with the average degree, with the largest bag (at the "center" of the TD) containing 77% of the nodes in the network in ER(32). Finally, the median density of the bags decreases dramatically as the density of the original graph increases. This initially counterintuitive phenomenon is easily explained: for extremely sparse ER, the TDs have many small bags, which need only small numbers of edges to have a reasonably high edge density. With the dense graphs, many nodes have to be placed in each bag, and this requires quadratically more edges per bag to achieve a similar density.

We would next like to look in more detail at the structure of the TDs generated on these different ER networks (e.g., what changes as we move from the central, large bags of the TD to the smaller, peripheral bags of the TD) as well as the internal structure of each bag. Recall, first, that, in a very sparse ER graph with expected degree greater than $2 \log 2$, but still sufficiently sparse, there are three different parts of the random network (two parts which may be viewed as core-like, one part which may be viewed as periphery-like) [83]. The core-like part of these graphs is biconnected, and it has an expander-like inner core (i.e., a set of nodes of higher degree), surrounded by an outer core, which has long chains of nodes (forming sparse cycles). The third, peripheral, part of the network consists of tree "whiskers" that hang off the biconnected core. A similar structure has been observed empirically when looking at low-conductance clusters/communities in a wide range of large social and information networks [77] and also when looking at the Gromov hyperbolicity and k-core properties of these real-world networks [6]. This contrasts sharply with the denser ER graphs, which are much more regular in terms of their degree variability, core structure, etc. Our results (here on extremely sparse ER graphs and below on PL graphs and many real-world graphs) demonstrate that TD heuristics can reflect this core-periphery structure.

5.1.2. Internal bag structure. Next, Figure 8 presents visualizations of three typical AMD bags for each of ER(1.6) and ER(32), respectively. In each case, the three bags are the most central (lowest eccentricity) bag in the TD (which we call the central bag), a typical bag that is a leaf in the TD (a periphery bag), and a typical bag that is in between these two in the TD (an intermediate bag). The color-coding is by k-core number, with high core nodes being red and low core nodes being blue. Note that the central bag for ER(1.6) is disconnected and consists of almost all singletons, whereas the central bag for ER(32) is well connected, and that the intermediate and peripheral bags for ER(1.6) are small, the latter consisting of only a single edge, whereas for ER(32) both the intermediate and the peripheral bag have nontrivial internal structure.

The increased density of ER(32) over ER(1.6) is the obvious cause of these differences, but it is worth considering what structures, produced by the increased density, affect the formation of the AMD TD. Recall from the toy graphs that heuristic TDs of cycles produced bags that had disconnected nodes. There were several different ways of producing the decomposition, but any TD of a small width on a cycle includes disconnected nodes in most bags. The more complex SMALLPLANAR has many small overlapping cycles. In

(a) ER(1.6) central bag subgraph.

(b) ER(1.6) intermediate bag subgraph.

(c) ER(1.6) peripheral bag subgraph.

(d) ER(32) central bag subgraph.

(e) ER(32) intermediate bag subgraph.

(f) ER(32) peripheral bag subgraph.

Figure 8 Bag subgraphs of an AMD TD of ER(1.6) and ER(32) graphs, colored by the k-core number of the node (red is high k, blue is low k). The central bag is the largest bag in the TD and one of the bags of minimum eccentricity; the peripheral bag is a leaf in the TD graph, and it achieves the minimum width in the TD; the intermediate bag is between these two extremes.

that case, the heuristics have to put many nonadjacent nodes into a bag. Essentially, cycles force distant nodes into the same bag, and many overlapping cycles will force many distant nodes into the same bag.

This intuition suggests (and we have confirmed by inspection) that a bag with many disconnected nodes, as in the central bag of ER(1.6) shown in Figure 8a, is due to a large number of overlapping cyclical structures. The intermediate bags of ER(1.6) contain nodes from the long, overlapping cycles of the outer core (and as these cycles do not overlap as much in periphery, these bags have fewer nodes), whereas the peripheral bags each contain a single edge, capturing the small trees on the periphery of the network (see also Figure 7). The coloring of the nodes indicates the core-periphery structure of the subgraph induced by the bags. In ER(1.6) there is only a 1-core (blue) and a 2-core (red), thus, the red nodes in the central bags are all in the 2-core, and the peripheral trees are in the 1-core, which agrees with [83].

To the contrary, in ER(32), whose core-periphery structure spans from a 7-core (blue) to a 23-core (red), although almost all of the nodes (94%) are in the 23-core, the central bag contains a relatively tightly connected mass of 77% of the nodes in the network. This begins to look more like SMALLER, which is a very dense ER network. The intermediate bags contain sparser structures (with some of the disconnected nodes and edges that are indicative of cyclical structures); and, although the peripheral bags still contain the smallest structures, in ER(32) they no longer contain only a single edge. This indicates that even the sparsest regions contain cycles and other complicated structures (but very few triangles, which agrees with the small clustering coefficient of these networks).

5.1.3. Large-scale organization. To provide a more quantitative evaluation of these ideas and to characterize better the large-scale organization of these synthetic networks, consider Figures 9, 10, and 11. These figures plot bag cardinality histograms, average bag density versus bag cardinality (this is width $+$ 1), and average k-core versus bag eccentricity for two ER networks (as well as a suite of PL and real-world networks). We will refer to other subfigures below, but for now consider only Figures 9a and 9e, Figures 10a and 10e, and Figures 11a and 11e for results on ER(1.6) and ER(32), respectively.

We saw in Figure 8a the central bag for ER(1.6), and we interpreted it in terms of the output of AMD TD as due to overlapping cycles; the histograms in Figure 9a show that for ER(1.6) (and the networks in the rest of Figures 9) there are only a very few such central bags. However, Figure 9 also shows that there are many very small bags in ER(1.6). Two features we noticed about these two types of bags are that there is a change
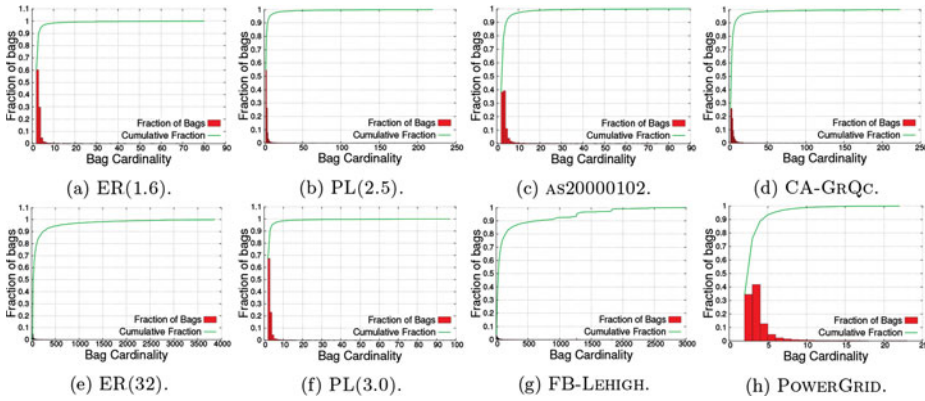
Figure 9 Bag cardinality histograms with cumulative fraction of bags for a representative set of networks. For all of the networks, there are many more small cardinality bags than large cardinality bags. This is consistent with a TD structure that has a few central bags that quickly taper and branch off into many small peripheral bags. As we will see in Figures 10 and 11, in networks with a strong core-periphery structure (e.g., not POWERGRID), these peripheral bags tend to have a low average k-core and high relative density. FB-LEHIGH has the most large bags, due to the tendency of the FB networks to form long, pathlike trunks in its TDs. POWERGRID has the smallest tapering effect; although the largest bags are still at the center of the decomposition, there is only a small change in size from the largest bags to the smallest, presumably because this network has the weakest core-periphery structure.

in edge density between the large and small bags and that there is a change in the k-cores represented between the large and small bags. Figure 10 and Figure 11, where the average edge density of a bag is plotted against the bag cardinality, show two ways of measuring this. These figures show that small peripheral bags are dense (relative to their small size—in the extreme case, this could be a single edge), and they contain low k-core nodes, as indicated, e.g., by the downward slope of the plots in Figure 11a.
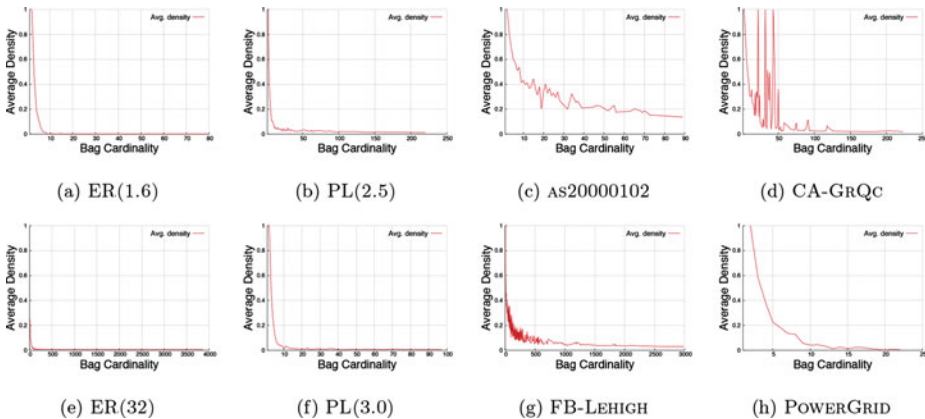
Figure 10 Average bag density versus bag cardinality plots for a representative set of networks. In the PL networks, the small bags are dense—in the extreme case consisting of a single edge—as in the ER(1.6) network; but the largest bags are larger and mostly connected, similar to the intermediate and central bags of ER(32). The real-world networks all show denser bags even at large-size scales compared to the synthetic networks, and this is due to the increased clustering present in these networks.
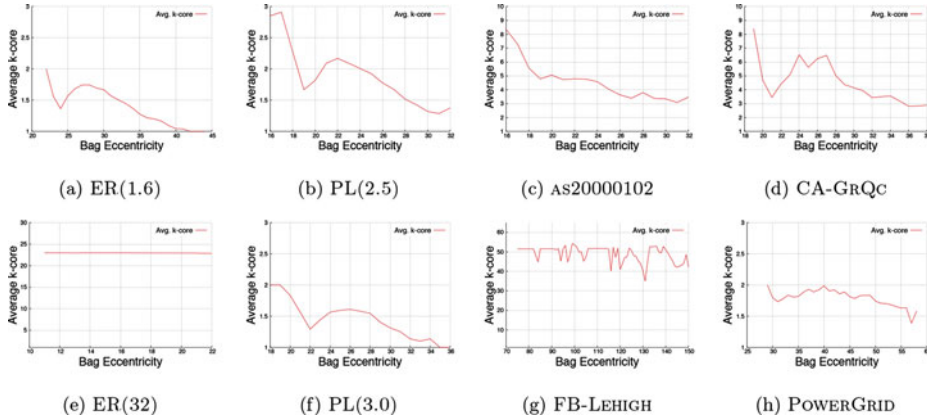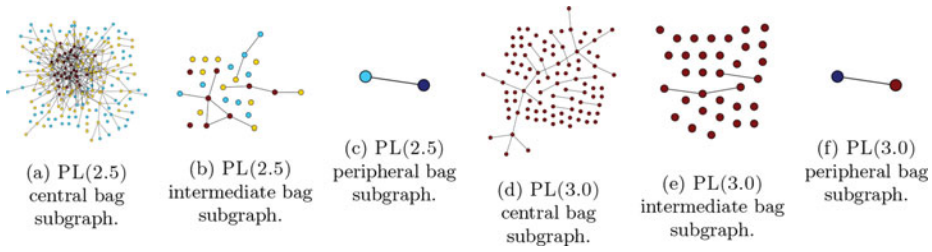
Figure 11 Average k-core versus bag eccentricity for a representative set of networks. The correlation between the core-periphery structure and the central-perimeter bags can be seen in a downward slope in these plots. Networks with no prominent core-periphery structure (ER(32) and POWERGRID, for two very different reasons) have a flat plot here; networks with moderate core-periphery structure (the PL graphs and ER(1.6)) have a downward sloping line, but relatively shallow (i.e., not deep) cores. AS20000102 and CA-GRQC both have prominent, deep, core-periphery structures that reveal themselves in this plot. The dips that show up at small eccentricities in several of the synthetic networks and CA-GRQC are due to the many small whiskers (in the sense of [77]) that hang off of the core bag. FB-LEHIGH also has a deep core-periphery structure (in the sense of k-core decompositions); but because of the long pathlike nature of the TD, and because most of the nodes are in the deepest cores, the plot is flat with larger downward dips as the bag eccentricity increases.

Many of the results for ER(32) are very different than those for ER(1.6). The histograms in Figure 9a show that this network has a much larger proportion of high-width bags than ER(1.6). The largely homogeneous core-periphery structure of dense ER networks should also be clear because the nodes, regardless of bag size, are mostly in the deepest core ($k = 23$). These trends can be seen by comparing the density of the smallest bags in ER(1.6) and ER(32) in Figures 10a and 10e. The flat plot of the average k-core in Figure 11e, which holds steady close to the value of the maximum k-core, indicates the lack of a core-periphery structure in the network.

Putting all of these results together, we can conclude that when it exists (e.g., in extremely sparse ER graphs), the core-periphery structure of ER networks is captured by the AMD TD; and when the core-periphery structure does not exist (e.g., for ER graphs for other even moderately sparse values of p), the large width of the TD indicates that most of the network is in the largest bag, which is analogous to most of the nodes being in the core of the network.

5.2. TDs of PL networks

Here, we give a summary of results of an analysis of TDs on PL random graphs, with an emphasis on the behavior as the degree heterogeneity parameter γ is varied. Recall that Table I provides basic statistics for the PL graphs. PL graphs are a class of ER-like random graphs, except that degree heterogeneity is exogenously specified. Previous work has shown that PL graphs have important similarities with extremely sparse ER graphs, when one is interested in small-scale versus large-scale treelike structure [77, 6, 65]. In particular, the increased-degree heterogeneity produces a large-scale core-periphery structure in the PL

(a) PL(2.5) central bag subgraph.   (b) PL(2.5) intermediate bag subgraph.   (c) PL(2.5) peripheral bag subgraph.   (d) PL(3.0) central bag subgraph.   (e) PL(3.0) intermediate bag subgraph.   (f) PL(3.0) peripheral bag subgraph.

Figure 12 PL(2.5) AND PL(3.0) bag subgraphs, colored by k-core number of the node.

networks, similar to the extremely sparse ER networks, but these PL networks also have some of the characteristics of denser ER networks (e.g., the core is more strongly connected and the diameter of the network is smaller).

We start with Table IVb, which shows the basic features of the TDs of PL networks. PL(3.0) has the least amount of degree heterogeneity and has similar characteristics to ER(1.6), and the lower degree exponents (PL(2.75), PL(2.5)) have characteristics similar to both the dense and sparse ER networks. Most notably, the maximum width increases (as it would if the density increased), and the median width and median bag density stay the same (low and high, respectively), as in the ER(1.6). In the previous section, we saw that the low median-width and high density were related to the presence of a core-periphery structure in the network. As we will see, this is also true of the PL networks, and the AMD TDs are again able to capture this structure.

Among other things, we find that for the PL networks, for a given average degree, the presence of very high degree nodes that tend to link to each other means that the density of the high-width bags (i.e., the core of the network, is greater), make it more like the cores of the denser ER networks. To the contrary, the peripheries of these PL networks are still very sparse, and TD bags including them look more like the peripheries of the sparse ER networks. The periphery results are reflected in the results presented in Table IVb, where we see that the median width is low and the median density is high (many bags with only a single edge within). The visualizations of the central, intermediate, and peripheral bags from TDs of PL networks in Figure 12 reflect this. In particular, the central bag for PL(2.5) looks somewhat like the intermediate ER(32) bags, whereas the central bag for PL(3.0) is much less well connected; and the peripheral bags for both PL graphs look like the ER(1.6) bags. The TD reflects the core-periphery structure via the central-peripheral bags, as reflected in the downward slope of Figure 11b.

As the power law exponent γ is increased, recall that the amount of degree heterogeneity in the resulting network is reduced, i.e., the number of high-degree nodes specified by the power law degree distribution is decreased. As a necessary consequence of maintaining this distribution as the nodes are connected, the high-degree nodes are likely to be connected to other high-degree nodes. This causes a core-periphery structure to emerge (see [6] for empirical measurements between the relationship between γ and the k-core structure).

For example, the core-periphery structure of PL(3.0) is shallower than that of PL(2.5), as seen in Figure 11. Similarly, the width of the PL(2.5) AMD TD is larger than that of the PL(3.0) AMD TD. In all cases, these widths are less than the corresponding ER(2) network, whereas one might expect these networks to have larger widths because of the increased core-periphery structure. This occurs because there are several factors to consider as γ is

decreased. The core *does* become denser as more edges are added to the core, causing these nodes to become more difficult to separate; but most of those extra edges come from the outer regions of the expander-like core, thus shrinking the size of the core and increasing the size of the periphery. In other words, when only a few medium degree nodes are added, then there are still cyclical structures in the core, as is observed in ER(1.6), except smaller; but as higher degree nodes are added, the core becomes denser and begins to become larger, as this forces larger and larger pieces of the core to be placed in the same bag, as is observed in ER(32).

The TD results on ER and PL graphs demonstrate that TDs (in particular, with the AMD heuristic) can capture the core-periphery structure of two common random network models. In both cases, to the extent that there was a core-periphery structure (which itself depended on the sparsity parameter p or the degree heterogeneity parameter γ), the central and peripheral bags in the TD from AMD were correlated with this structure. The peripheral bags were smaller and much sparser than the central bags and contained nodes from the shallow (low) k-cores of the network. With the exception of the treelike periphery of the extremely sparse ER and the PL networks, the structures observed in TDs of the random network models were largely driven by loosely connected core structures (e.g., overlapping loops in sparser regions and expander-like cores in the denser regions). This is consistent both with the results on the toy networks and previous work involving the structure of ER networks [50, 83]. (The one exception to this is the denser ER(32), where the central bag contained 77% of the network; in this case, the network does not exhibit the core-periphery structure of the other networks looked at in this section.) We will see how we obtain similar results when applying the AMD heuristic to real-world networks.

6. TREE DECOMPOSITIONS OF REAL-WORLD NETWORKS

In this section, we will describe the results of using a variety of TD heuristics on a set of real-world networks. Our goal is to use the insights from the previous sections to evaluate the performance of existing TD heuristics on real social and information networks and to understand how those TDs can be used to obtain an improved understanding of the properties of these realistic networks. Our main results in this section are three-fold. First, in Section 6.1, we summarize results of a detailed empirical evaluation of the AMD TD heuristic applied to our suite of realistic networks.[13] The main focus is to illustrate how these TDs capture previously identified core-periphery structure, and also to illustrate how the internal structure of TD bags can be understood in terms of large-scale cycles and small-scale clustering in the original graph. Second, in Section 6.2, we evaluate the ability of AMD to identify small-scale good-conductance communities such as those previously identified by the NCP with local spectral methods [77, 65]. We show connections between bags that are more peripheral in the TD and small good-conductance communities responsible for dips in the NCP. Third, in Section 6.3, we illustrate that TD heuristics can be used to identify certain other types of large-scale nonconductance-based ground truth communities. In particular,

---

[13]We should note that we ran these computations with many different TD heuristics. In most of this section, however, we show results from only (the most scalable) AMD heuristic. This is simply for brevity. There were some differences from heuristic to heuristic, but we feel this one is representative of the type of behavior found.

we will show connections between bags that are more central in the TD and large-scale communitylike (by a ground truth metric but not by conductance quality) clusters.

6.1. Results on identifying core-periphery structure

Here, we will describe the results of an empirical evaluation of the AMD TD heuristic applied to our suite of real-world networks (those in Table I). We will begin in Table V with a brief survey of all of our real networks, and we will then focus on four representative networks: AS20000102, CA-GRQC, FB-LEHIGH, and POWERGRID. The first three all exhibit some form of previously recognized core-periphery structure [6], whereas POWERGRID is known to lack a strong core-periphery structure (basically because it is heavily tied to the underlying locally Euclidean geometry of Earth [6]). Note, though, that the Facebook networks are very core heavy in the sense that they have many nodes in deep cores, essentially because of their significantly higher average degree (see, e.g., [6] and Table I). (Thus, informed by previous results on k-core decompositions and related treelike techniques [6, 94, 77, 45], we expect to see evidence of the core-periphery structure in the TDs associated with AS20000102, CA-GRQC, and, to a lesser extent, FB-LEHIGH, but a lack of substantial core-periphery structure in the TD of POWERGRID.)

6.1.1. Overview of core-periphery results for all networks. In Table V, we present the number of bags in the AMD TD (N_{AMD}), the maximum eccentricity (diameter) of the TD (E_{AMD}), the maximum and median width of the TD (W and $\tilde{W}$, respectively), and median bag density ($\tilde{D}$). These measurements provide us with an idea of how large the most connected part of the network (maximum width) is and how numerous small bags are (median width), which is indicative of areas of the network that have small separators, in our case small peripheral regions of the network; and whether the small separators are more cliquelike or consist of mostly disjoint nodes (median density), with disjoint nodes being indicative of cycles and cliquelike structures being indicative of more meaningful communities. A large maximum width combined with a low median width is evidence a deep core and a shallow periphery, and high median density is evidence of a periphery based on more communitylike separators, rather than more disparate separators. These observations assume that high-density bags are mostly small-width bags. This assumption is plausible, given that as the width w of a bag increases, the number of edges required to maintain a constant density increases as w^2; and in many cases, we have confirmed this assumption indirectly or by direct observation. For example, see our discussion of bag density and bag width following, as well as Figure 10 for empirical evidence that high-density bags are generally the smallest-width bags.

Networks based on an underlying Euclidean geometry (e.g., ROAD-TX, POWERGRID) have low maximum widths and low median widths, which indicates that they do *not* have a strong core-periphery structure. Although these networks have many small-width bags, which is indicative of treelike sections of the network, the internal subgraphs have a low median density (e.g., as compared to certain ER networks). More social networks, such as POLBLOGS and the Facebook networks, all have higher average degrees and, consequently, higher widths, with lower median widths. This is one indicator of a core-periphery structure. Because the median widths are higher in these networks, compared to the other real networks (although lower than the ER networks), the peripheral structure in these social graphs tends to be denser than that in the other networks. Also, the median density, although low compared to the other real networks, is very high compared to the median density of the densest ER networks in Table IVa. Thus, although the periphery is more difficult to separate

Network	N_{AMD}	E_{AMD}	W	$\tilde{W}$	$\tilde{D}$
CA-GRQC	3014	39	222	2	1.0
CA-ASTROPH	10708	78	3616	5	1.0
AS20000102	6364	33	88	2	1.0
GNUTELLA	6475	33	1629	2	0.67
EMAIL-ENRON	26781	78	2237	3	1.0
FB-CALTECH	395	30	357	18	0.53
FB-HAVERFORD	516	56	891	37.5	0.38
FB-LEHIGH	1919	151	2983	31	0.32
FB-RICE	1481	76	2553	31	0.37
FB-STANFORD	4809	100	6674	16	0.38
POWERGRID	4666	59	21	2	0.67
POLBLOGS	899	49	294	6	0.57
ROAD-TX	1.25×10^6	170	197	3	0.5
WEB-STANFORD	2.10×10^5	500	1419	5	0.83

Table V Statistics for TDs of real networks. Notation is the same as that in Table IV.

into small good-conductance, communitylike clusters than some of the other real networks, e.g., CA-GRQC or CA-ASTROPH, it is still formed from more communitylike pieces than similar ER (or PL) networks.

Observe also that the two web networks, GNUTELLA and WEB-STANFORD, have high widths, similar to the Facebook networks, but lower median widths; and that they have higher median densities (especially when compared to ER(4), ER(8), and ER(16)). This indicates that these networks have a sparser, more treelike periphery than other social networks, which is also consistent with previous results [77]. Importantly, and also consistent with previous results [77], is that the sparsest, most treelike peripheries belong to the collaboration, email, and autonomous systems networks (CA-GRQC, CA-ASTROPH, EMAIL-ENRON, AS20000102). These networks all have low median widths, high median densities, and high maximum widths, indicating that they exhibit the cleanest core-periphery structure, also consistent with the upward-sloping NCPs [77, 65].

6.1.2. More details on core-periphery structure of four representative networks. We will now look at several representative networks in greater detail. Let us start by discussing Figures 9, 10, and 11 from Section 5. Figure 9 clearly shows that most of the bags in the TDs are small-width bags. In fact, proportionally, FB-LEHIGH has the largest fraction of large bags, and yet 80% of the bags are below width 200 (in a graph with 5073 nodes, where the TD has a maximum width bag of 2983). Since the bags and edges of TDs form separators in the network, this indicates that there are many relatively small separators. These are the largest in Facebook networks, where the separators tend to have around 100 nodes, whereas in most other networks many of the separators have around 10 nodes. This is consistent with typical views of core-periphery structure, with a few more highly connected nodes in the core and many less well-connected nodes in the periphery. That is, in order to separate most pieces of the periphery (where "piece" is defined by the end of branches in the TD), only 10 or fewer nodes are needed for most
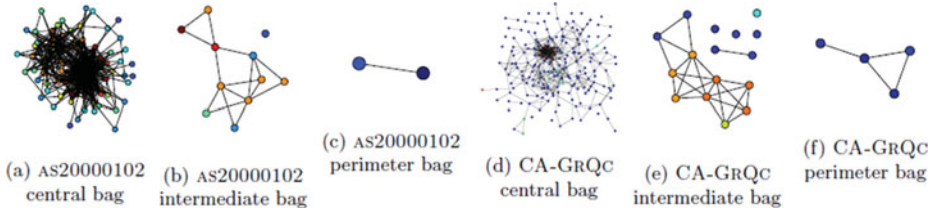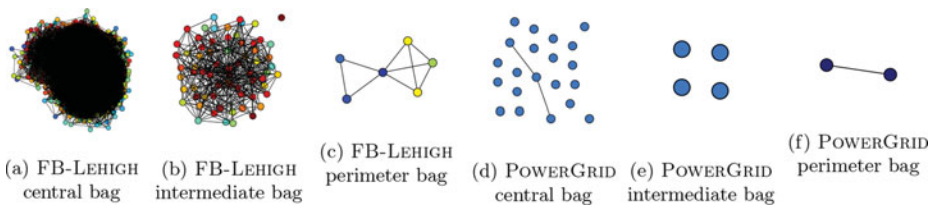
Figure 13 AS20000102 and CA-GRQC AMD bag subgraphs, colored by *k*-core number, with red indicating deep/high *k*-cores and blue indicating shallow/low *k*-cores.

of the social/information networks, although ca. 100 nodes are needed for the Facebook networks.

In Figure 10, we see the average edge density for bags of a given cardinality plotted against the bag cardinality, showing that small-width bags have high densities. An important distinguishing feature of the three representative real networks (that are not tied to an underlying Euclidean geometry) is that the curve has a heavier tail than that in the synthetic networks. This indicates that separators, up to much larger size scales, are less disparate (e.g., are denser or clumpier) than in the synthetic networks. In POWERGRID, on the other hand, the underlying Euclidean geometry leads the density to fall off more quickly. It falls off similarly to the sparse ER network, except that the tail of the curve is shorter. In this case, only the smallest bags have tight separators.
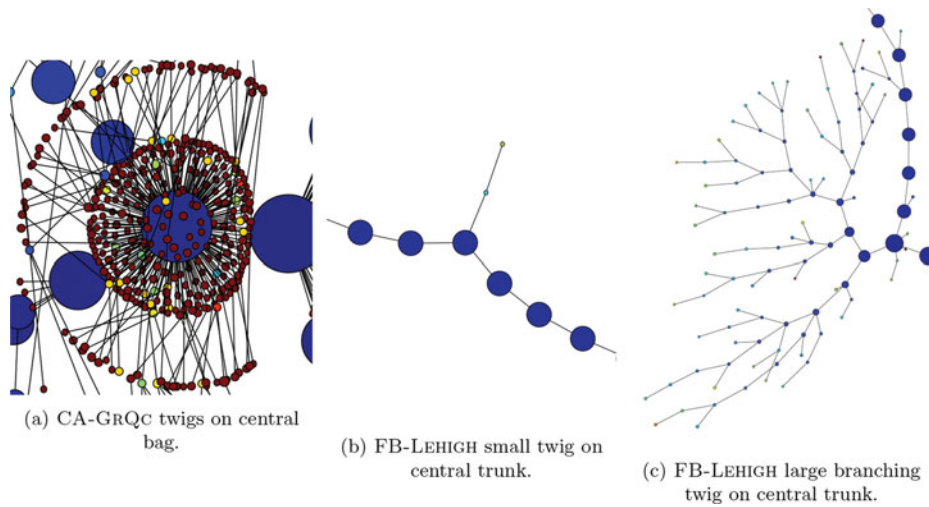
In Figure 11, we consider the relationship between the core structure and low eccentricity (central) bags, and we compare that with the relationship between the periphery structure and the high eccentricity (perimeter) bags in the TD. Figures 11c and 11d show that for AS20000102 and CA-GRQC there is a clear downward trend as the bag eccentricity is increased. This indicates that low eccentricity bags contain more high *k*-core nodes on average and that the high eccentricity bags contain more low *k*-core nodes on average. Figure 11g shows that FB-LEHIGH, due to its greater density, has a mostly flat profile until the most extreme reaches of eccentricity are met, at which point some of the bags begin to contain nodes of a lower *k*-core. Thus, the core-periphery structure is present in FB-LEHIGH, but the core-periphery structure is moderated by a very large core which produces long pathlike sets of nodes that, in turn, lead to large core bags and, hence, a much larger eccentricity. (This is typical of the results for most of the Facebook networks, which is consistent with their flat NCP [65].) Finally, POWERGRID, which is not expected to have exhibited a correlation between *k*-core structure and bag eccentricity, has a flat profile.

To illustrate these findings, we present visualizations in Figures 13 and 14. Shown are a central or very deep core bag, a perimeter or very peripheral bag, and an intermediate

Figure 14 FB-LEHIGH and POWERGRID AMD bag subgraphs, colored by *k*-core number, with red indicating deep/high *k*-cores and blue indicating shallow/low *k*-cores.

(a) CA-GRQc twigs on central bag.

(b) FB-LEHIGH small twig on central trunk.

(c) FB-LEHIGH large branching twig on central trunk.

Figure 15 Twigs on FB-LEHIGH and CA-GRQc. Bags are colored by the density, blue indicating low density and red indicating high density. A small twig and a larger, branching twig on the FB-LEHIGH trunk are shown. These twigs, combined with the long, pathlike trunk, cause dips in the k-core eccentricity plot. In CA-GRQc, the concentration of the twigs on one or two central bags causes only a single, large dip compared to the multiple dips in FB-LEHIGH. Synthetic networks in Figure 11, (PL(2.5), PL(3.0), and ER(1.6)) also have twigs similar to CA-GRQc.

bag, for each of our four networks. These figures show the communitylike nature of typical bags for the three information networks, i.e., AS20000102, CA-GRQc, and FB-LEHIGH, as well as the more disparate separators of the POWERGRID. The coloring of the visualizations in these figures is by k-core: the red nodes are in deep (high) k-cores and the blue nodes are in shallow (low) k-cores.

One final observation we would like to make is to address the "dips" in the average k-core curves shown in Figure 11 (e.g., the dip in Figure 11d at a bag eccentricity of 21 or in Figure 11g throughout). These dips are due to what we will call "twigs," where a twig is a small (low width and short) branch off of a much larger (high width and long) trunklike structure of the TD. For example, in FB-LEHIGH and in the other Facebook networks, the high average degree results not only in high widths, but in larger collections of bags of high width. These are arranged in a long path (a "trunk") with many branches at either end. Along this main trunk, there are occasional twigs, which contain peripheral nodes. Because the trunk is long, the average k-core at the point where the twig is attached is slightly lower, resulting in the dip in the curve. In Figure 15, we provide a visualization of the twigs responsible for three of these dips.

6.1.3. Summary of large-scale core-periphery and treelike structure in real-world networks. These empirical observations suggest that many realistic social/information networks have a nontrivial core-periphery structure; and that in many cases this is caused by many small overlapping clusterlike or moderately cliquelike structures. That is, there is local nontreelike (combinatorial) structure that "fits together" into a global core-periphery structure that is treelike (in a metric and/or cut sense) when viewed from large size scales. This is in sharp contrast with many models and intuitions. Most obviously, this is in contrast with the random networks (in particular, the not extremely sparse ER networks and to a lesser extent the PL networks, but many other more popular

random generative models), which have a locally treelike, but globally loopy structure. Less obviously, this is also in sharp contrast with networks such as POWERGRID, PLANARGRID, and ROAD-TX that are strongly tied to an underlying Euclidean geometry. Said another way, many realistic social/information networks have a more tightly connected corelike structure than is present in typical random networks, and they have peripheral and intermediate regions that are "clumpier" than these random networks. Although these claims are perhaps intuitive, our empirical observations demonstrate that they can be meaningfully identified with TDs and interpreted as leading to large-scale cut-based treelike structures.

Interestingly, aside from the local clumpiness, the real-world social/information networks do have a core-periphery structure that is reminiscent of that which is also seen in extremely sparse ER graphs and PL graphs with greater degree heterogeneity. (This, too, is consistent with prior results suggesting that extreme sparsity coupled with randomness/noise is responsible for the dips in the NCP [77, 65].) It is also worth emphasizing that in most of the intermediate bags of the real social/information networks, there are still a small number of disconnected nodes. This indicates that there are still a small number of alternate paths, which are disparate from the clusters, to the rest of the nodes in the network. The most prominent exceptions to these general observations are networks that either do not have a strong core-periphery structure, e.g., POWERGRID that is tied to a two-dimensional underlying Euclidean geometry, or networks that have a relatively low clustering coefficient, e.g., Gnutella09. In both of these cases (but for different reasons), the internal subgraphs of the intermediate and peripheral bags have a larger number of disconnected nodes than the other realistic networks.

## 6.2. Connections with good-conductance communities results

Here, we will consider how the peripheral part of the treelike core-periphery structure identified by TDs relates to low-conductance clusters/communities that were previously identified by the NCP method [77, 65]. To do so, observe that one way to determine whether a TD captures clustering/community structure is to see if those clusters/communities are well localized in the TD. By "well localized," we mean that the cluster/community is contained in a relatively small number of (contiguous) bags. We followed previous personalized page rank (PPR) local spectral procedures [12] to generate a set of candidate clusters [77, 65]. Then, given a set of candidate clusters, we looked at how many bags in the TD contain at least one node from this cluster, i.e., we measured how well localized the community is in the TD. As a crude threshold of whether a cluster/community is localized, we consider it to be localized if it is contained in fewer bags than there are nodes in the community.[14] We apply this method using the AMD heuristic.

Our results for several real-world and synthetic networks are presented in Figures 16–20. For each figure/subfigure, the horizontal axis represents community size in number of nodes (on log scale), and the vertical axis is either the conductance of the best community found using the PPR method (recall that a low conductance represents a better community)
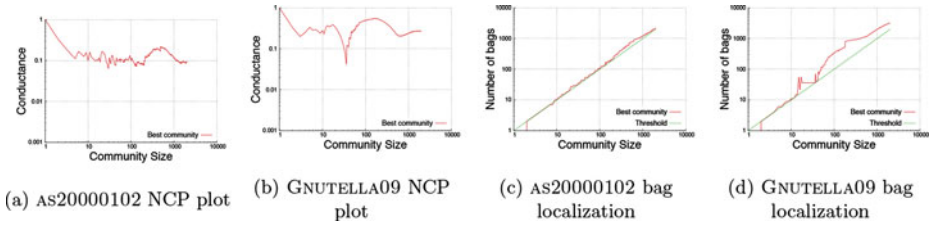
---

[14]To understand this threshold, consider the following example: if a community of size n is a tree, e.g., whiskers in ER(1.6), then it will be contained in n bags in the (ideal) TD; if the community is a "clique whisker," i.e., a clique connected to the rest of the network by only one edge, it will be contained in just one or two bags; and if the community contains deep core nodes, which are connected to many nodes outside of the community, the community will be spread across many bags in the network. Other measures of TD locality showed similar results.
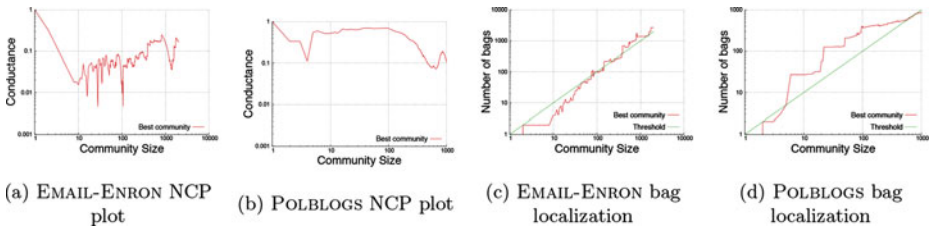
(a) ER(1.6) NCP plot   (b) ER(32) NCP plot   (c) ER(1.6) bag localization   (d) ER(32) bag localization
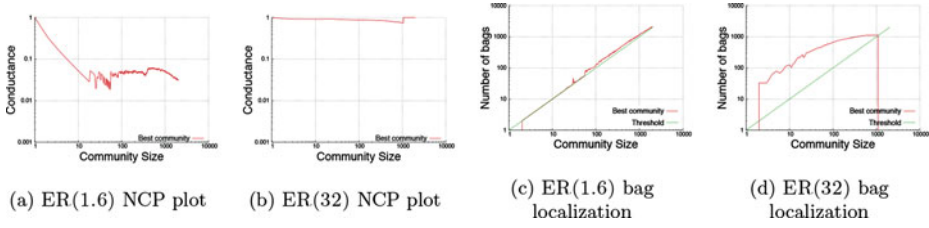
Figure 16 ER(1.6 and ER(32) NCP plots and tree-localization plots. The localization threshold is plotted in green.

(a) CA-GRQC NCP plot   (b) FB-LEHIGH NCP plot   (c) CA-GRQC bag localization   (d) FB-LEHIGH bag localization
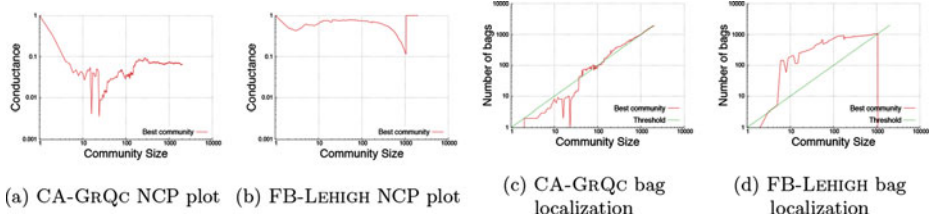
Figure 17 CA-GRQC and FB-LEHIGH NCP plots and tree-localization plots. The localization threshold is plotted in green.

(a) AS20000102 NCP plot   (b) GNUTELLA09 NCP plot   (c) AS20000102 bag localization   (d) GNUTELLA09 bag localization

Figure 18 AS20000102 and GNUTELLA09 NCP plots and tree-localization plots. The localization threshold is plotted in green.

(a) EMAIL-ENRON NCP plot   (b) POLBLOGS NCP plot   (c) EMAIL-ENRON bag localization   (d) POLBLOGS bag localization

Figure 19 EMAIL-ENRON and POLBLOGS NCP plots and tree-localization plots. The localization threshold is plotted in green.

or the number of bags that contain members of the community (again, on log scale). In the bag plots, the red line represents the number of bags that contain a node *from the community in the corresponding NCP plot*, and the green dashed line represents the locality threshold. When the number of bags for a given community is localized by our definition, the red plot will be below the green threshold.

(a) PLANAR NCP plot
(b) POWERGRID NCP plot
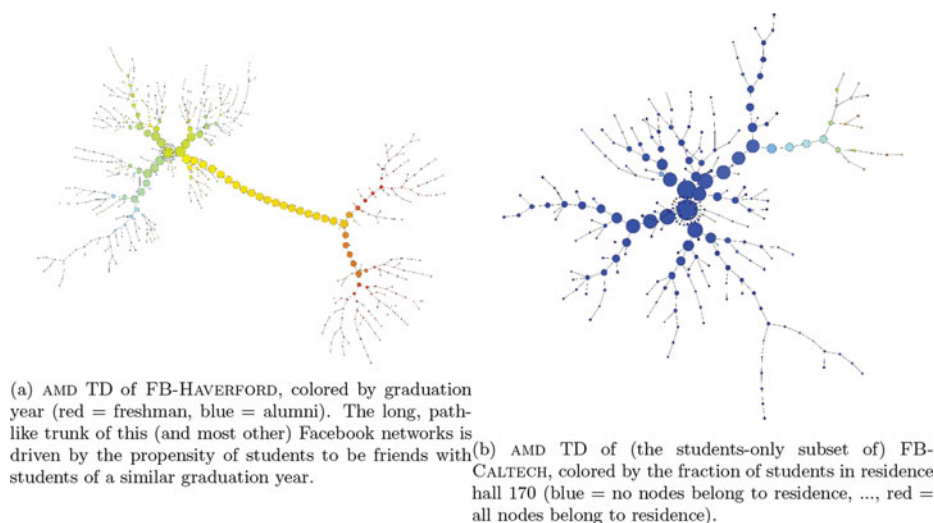(c) PLANAR bag localization
(d) POWERGRID bag localization
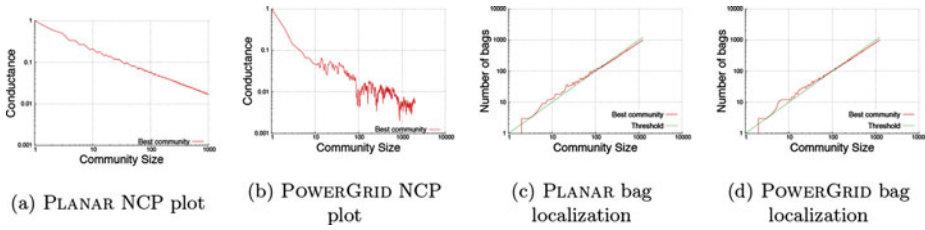
Figure 20 PLANAR and POWERGRID NCP plots and tree-localization plots. The localization threshold is plotted in green.

As a reference, consider the extremely sparse and somewhat denser ER networks, which are shown in Figure 16. Because it is so sparse, ER(1.6) does have some very small good-conductance clusters. As shown in the figure, however, the small "communities" are contained in roughly the same number of bags as there are in the community. This is expected, because these communities are largely peripheral treelike whiskers in the network (Section 5). For larger communities, which include core nodes, the localization is slightly above the line defining our threshold. To the contrary, for the denser ER(32), there are no good-conductance clusters at any size, and bag localization is above the line defining the localization threshold, indicating that the localization is poor at all size scales.

For the small and intermediate-sized clusters in many of the real networks (including many of those from [77]), the smaller good-conductance clusters found using the PPR method are reasonably well localized within the TD, whereas the larger, poorer-conductance clusters are not. Consider, e.g., CA-GRQC in Figure 17 as an example. However, both large and small clusters found with the PPR method applied to the denser graphs from the FACEBOOK100 set (i.e., those that do not have even small-cardinality good-conductance clusters [65]) are not well localized in the TD. Consider, e.g., FB-LEHIGH in Figure 17 as an example. Figure 18 shows AS20000102 and GNUTELLA09, which also shows NCP plots that do not yield small good conductance clusters, and which shows that the outputs of the PPR method are not particularly well localized in the TD. Figure 19 shows that EMAIL-ENRON does have some of its small good-conductance clusters well localized, and it also shows that the output of the PPR algorithm applied to POLBLOGS leads to medium-to-large clusters with poor conductance values that are poorly localized in the TD.

Finally, although networks with an underlying Euclidean geometry are of less interest for social/information network applications, for completeness, it is worth considering how these TD methods apply to them. Figure 20 presents results for PLANAR and POWERGRID. Both of these networks have downward-sloping NCP plots, which are different from the other social and information networks, reflecting the Euclidean geometry underlying these networks. In both cases, fairly uninteresting results are obtained, suggesting that the localization metric we propose is more interesting for realistic social graphs with nontrivial treelike core-periphery structure.

Although our results demonstrate that good-conductance clusters/communities in several realistic social graphs are well-localized in TDs found with existing heuristics, it is not obvious how to address the reverse question of finding good-conductance communities from a TD. One could attempt to look at all or some large number of combinations of bags in the TD. Because one is usually interested in well-connected communities/clusters, the running intersection property of TDs could be used to restrict attention to connected subsets of a TD. There are, however, two obvious issues. First, there does not exist an

(a) AMD TD of FB-HAVERFORD, colored by graduation year (red = freshman, blue = alumni). The long, path-like trunk of this (and most other) Facebook networks is driven by the propensity of students to be friends with students of a similar graduation year.

(b) AMD TD of (the students-only subset of) FB-CALTECH, colored by the fraction of students in residence hall 170 (blue = no nodes belong to residence, ..., red = all nodes belong to residence).

Figure 21 AMD TD of FB-HAVERFORD and FB-CALTECH. FB-HAVERFORD is presented, rather than FB-LEHIGH (which has similar large-scale TD structure), because its smaller eccentricity (56 rather than 150) makes it easier to visualize. For FB-CALTECH, this is a graphical representation of data presented in Table VII for the AMD TD and residence hall 170.

obvious analogue of the "sweep cut" used in the spectral partitioning method for finding the best community from a TD. Second, as a related practical matter, the presence of high-degree (or deep core) nodes in the intermediate and central bags of a TD cause bags to be poor-conductance communities. These nodes have many connections and increase the "surface area" of most cuts, even if there is only a small number of them in a cluster. We observed that, in the clusters we found, using the PPR method, each cluster is typically well represented by a set of small bags plus a couple of nodes in the larger bags. If we then attempt to form clusters by combining bags, we get *all* of the nodes in the larger bags, including deep core nodes. Additional methods of filtering nodes for the larger bags, such as ordering by node degree or k-core combined with a sweep cut, might improve these results.

6.3. Results on identifying ground-truth communities

Here, we will consider other ways in which the output of TDs can be useful in identifying clusters/communities of interest to the domain analyst. In particular, we describe two examples from the demographic data associated with the Facebook100 dataset [100].

Consider, first, Figure 21a, where we show the AMD TD of the FB-HAVERFORD network, and where each bag is colored coded by the average graduation year of the constituent nodes. There is a large linear or trunklike structure that dominates the large-scale structure of the TD. We observe that there is a strong overlap between the nodes that comprise successive bags in that trunk, and we note that this trunklike structure is typical of most of the FACEBOOK100 networks (but is not seen in most other social graphs we have considered). Also, each end of the long trunk correlates strongly with graduation year, and there is a gradual change in the average graduation year of each bag as we move across the trunk. Thus, to the extent that one accepts graduation year as some sort of easily

quantifiable ground-truth community, the large bags in the TD of this network seem to be capturing a legitimate ground-truth structure in the network. This fits well with prior results that report that, in most of the Facebook networks, graduation year is the best predictor of the existence of edges between two nodes [100].

Consider, next, Figure 21b. It is known that for a small number of the FACEBOOK100 networks (e.g., FB-CALTECH, FB-RICE, and FB-UCSC), residence hall rather than class year is the best edge predictor [100]. Thus, we considered the AMD TD of (the students-only subset of) FB-CALTECH. In this case, a single simple trunklike structure is not dominant, but there are several relatively large peripheral branches, and many of the peripheral branches are dominated by a particular residence hall. In Figure 21b, the bags are colored by the fraction of students in residence hall 170 (chosen arbitrarily). These examples are of particular interest because good-conductance clusters do not exist in FACEBOOK100 networks [65].

By looking at bags where the concentration of a particular community node is higher than the incidence of that community throughout the TD, we can form a very simple classification rule. In particular, given residence hall X, we collected all bags whose fraction of nodes, which were listed as belonging to residence X, was higher than the fraction of nodes belonging to that hall in the network (the incidence in the network is given in column F in Tables VI and VII). We then kept the largest contiguous set of bags and used membership in this set as the classifier. Although this is an overly simple classifier, the goal in this section is simply to provide a baseline for how the residence communities are located in the TD.

We performed this procedure on the students-only restriction of FB-CALTECH. Tables VI and VII provide a summary of the classification results using this method on FB-CALTECH network, with the (anonymized) listed residence hall for that student as the community. Table VI shows the fraction of the ground-truth community captured by the largest contiguous set of bags described. This is analogous to the recall of classifying the community using this branch in the TD. Table VII shows the fraction of the nodes

Hall	F	MINDEG	MINFILL	LEXM	MCS	AMD	METNND
None	.134	.270	.257	.270	.324	.284	.297
165	.066	.472	.528	.556	.528	.472	**.861**
166	.090	.736	.736	**.925**	.811	.642	.792
167	.134	**.642**	.566	.453	.585	.491	.566
168	.116	.746	.762	**.952**	.889	.825	.143
169	.136	.726	.712	**.904**	.877	.658	.740
170	.090	.725	.725	.739	.783	**.855**	.362
171	.136	.714	.673	.776	**.857**	.592	.429
172	.098	.630	.630	.534	.699	.548	**.863**

Table VI Fraction of each FB-CALTECH residence hall captured in the largest contiguous set of "frequent" bags. A frequent bag is a bag where the fraction of students who belong to the given residence hall is greater than the fraction of students who belong to that residence in the entire network. Column F gives the fraction of students who identified as being in the associated hall (i.e., the threshold for being a frequent bag for that residence hall). This procedure was also performed for the nodes that did not have a residence hall listed for comparison.

Hall	F	MINDEG	MINFILL	LEXM	MCS	AMD	METNND
None	.134	.065	.064	.068	.071	.071	.100
165	.066	.055	.057	.052	.052	.059	**.086**
166	.090	.104	.110	.111	.111	.092	**.129**
167	.134	**.102**	.097	.092	.087	.088	.092
168	.116	.137	.140	.150	.147	**.157**	.043
169	.136	.150	.151	.147	.163	.138	**.187**
170	.090	.134	.139	.145	.140	**.171**	.103
171	.136	**.105**	.099	.100	.104	.084	.074
172	.098	.131	.129	.100	.137	.115	**.199**

Table VII Fraction of the nodes contained in the largest contiguous set of frequent bags for a given residence hall which *actually belong* to the given residence hall. A frequent bag is a bag where the fraction of students who belong to the given residence hall is greater than the fraction of students who belong to that residence in the entire network. Column F gives the fraction of students who identified as being in the associated hall (i.e., the threshold for being a frequent bag for that residence hall). This procedure was also performed for the nodes that did not have a residence hall listed for comparison.

in the union of all bags in this largest contiguous set, which belong to the community. This is analogous to the precision of classifying the community using this branch. Since FB-CALTECH is very small, we can use a much larger variety of TD as classifiers than is possible for larger networks, and we present results for all of these TD classifiers.

Although the communities do seem to be well captured by the TDs, there are also many other nodes in the same bags as these communities (see Table VII). Although the only bags selected were bags where the residence hall in question was overrepresented, combining these bags actually resulted in a lower concentration of residents than were present in the network for some residence halls (see Table VII where the values are lower the F for a given residence hall). This occurs because the nonresident nodes in each of these bags are different, and the resident nodes are largely the same for each bag in the branch.

In terms of heuristic performance, the MINDEG, MINFILL, and AMD seem to have similar performance given a residence, although there seems to be a larger gap between AMD and the other heuristics. This is not surprising because these are all greedy heuristics that work by reducing fill (or minimum degree, which is a proxy for fill) in each step. LEXM and MCS also seem to behave similarly, and they have the best performance in terms of recall (Table VI). METNND has a different profile from the other networks and seems to do the best in terms of precision (Table VII). These results are comparable to what can be obtained with other simple classification rules, and they suggest that TDs could be useful in these types of machine learning applications.

Overall, these results demonstrate that for these realistic social/information networks, several types of plausible ground-truth communities are well correlated with the large-scale structure identified by existing TD heuristics. This is striking because these heuristics make local greedy decisions about how to form the TDs, and it suggests that improved results could be obtained in this application by considering TD heuristics designed for graphs with this type of structure.

7. MORE DETAILS ON TREE DECOMPOSITION METHODS

In this section, we consider the question of whether TDs and their treewidths can be related to other parameters for treelike structures, specifically the Gromov δ hyperbolicity.[15] It might appear that there is no relation between TDs and δ (since, e.g., treewidth and δ take on opposite extremal values on cliques and cycles), but there are, in fact, structural characterizations for when they align. We will present here our new theoretical results on relating TDs and δ-hyperbolicity. Although this result is a relatively straightforward extension of previous work [46], and although most of the rest of the article can be understood without this result, we include it here for completeness: first, since motivating prior work in [6] demonstrates an empirical connection between the cut-based treelike notion from TDs and the metric-based treelike notion from δ-hyperbolicity; and second, because our results in Section 6 demonstrate the inadequacy of a naïve optimization of treewidth and the importance of large cycles for realistic social graphs.

7.1. Treewidth, treelength, and hyperbolicity

We start with the following definition, which provides another quality measure of a TD [47]. See also [49].

Definition 7.1. Let $\mathcal{T} = (\{X_i\}, T = (I, F))$ be a tree decomposition of a graph G. The *length* of $\mathcal{T}$ is defined to be $\max_{i \in I, x, y \in X_i} d_G(x, y)$, where $d_G(x, y)$ is the shortest path distance in G. Analogously to treewidth, the *treelength* of G, denoted $tl(G)$, is the minimum length achieved by any tree decomposition of G.

It is straightforward to see that the treelength is at most the diameter of G. As with treewidth, finding a tree decomposition achieving minimum length (and in fact the treelength itself) is NP-hard [79]. Given this, one might ask whether treelength and treewidth can be simultaneously approximated. For general graphs, [47] proved a negative result.

Theorem 7.2. *[47] . Any algorithm computing a tree decomposition approximating the treewidth (or the treelength) of an n-vertex graph by a factor α or less does not give an α-approximation of the treelength (resp. the treewidth) unless $\alpha = \Omega(n^{1/5})$.*

The specific examples used by [47] to prove their negative result are modifications of the 2-dimensional mesh (i.e., a lattice), which—due to long induced cycles—is not δ-hyperbolic for small values of δ. This suggests that the situation might be very different for "real-world" graphs—which have small diameter and which have nontrivial embedding properties into low-dimensional hyperbolic spaces. (This is an open area of research more generally.) It was shown [40] that if $tl(G) \leq \lambda$, then G is λ-hyperbolic, and that a δ-hyperbolic graph G on n vertices satisfies $tl(G) \leq 17 + 12\delta + 8\delta \log_2 n$. Unfortunately, for many real networks of interest, this is *not* an improvement on the trivial bound of diameter, because their diameter alone will be less than $O(\log_2 n)$. We conjecture that under minimal additional conditions, a δ-hyperbolic graph with diameter D has treelength at most a function of $\log_2 D$, a vast improvement on both known bounds.

---

[15]As we mentioned in Section 2, this is not the main focus of our work, but there has been recent theoretical and empirical interest in this and related questions; see, e.g., [35, 105, 47, 79, 58, 72, 48, 2, 1].

We turn to the question of using additional structural properties to characterize the interplay between δ, $tw(G)$, and $tl(G)$. The following theorem is our main result; this theorem follows from the work of Müller on atomic TDs [46], and its proof is in Section 7.2.

Theorem 7.3. *[86]. Say a subgraph H of G is geodesic if $d_H(u, v) = d_G(u, v)$ for all $u, v \in V(H)$. Let $v(G)$ be the length of a longest geodesic cycle in G. Then,*

$$\delta(G) \leq tl(G) \leq (tw(G) + 1) \cdot v(G).$$

Further, this result is tight—there is a graph class $\mathcal{G}$ of unbounded treewidth and containing arbitrarily long geodesic cycles such that $\delta(G) = \Theta(tw(G) \cdot v(G))$ for every graph $G \in \mathcal{G}$.

In other words, if we can eliminate long distance-preserving cycles and obstructions to low treewidth (large grid minors), then G will embed well in low-dimensional hyperbolic space.

7.2. Proof of Theorem 7.3

Before we can give the proof of Theorem 7.3, we need a few additional definitions. First, given a rooted tree T and a node $s \in T$, define T_s to be the subtree of T with root s:

$$T_s := T[\{t \in T \mid s \text{ is an ancestor of } t\}].$$

For a graph $G = (V, E)$ with tree decomposition $(\{X_i\}, T)$, where T is rooted arbitrarily, for $s \in T$ define $G_s := G[\bigcup_{t \in T_s} X_t]$ to be the graph induced by those bags that are equal to or below X_s in the decomposition. We will write $N(S)$ for the neighbors of a set S — more precisely, $N(S) = \{u \in V \mid (u, s) \in E \text{ for some } s \in S\} \setminus S$. Finally, for notational convenience, for $x \in V$ and $e \in E$, we will write $G - x$ for the graph $(V \setminus \{x\}, E)$ and $G - e$ for the graph $(V, E \setminus \{e\})$. We now define a special type of tree decomposition (so-called *atomic tree decompositions*), and give a crucial property of all vertices that co-occur in one of its bags.

Definition 7.4. [Atomic Tree Decomposition, [19]]. Let G be a graph on n vertices. The *fatness* of a tree decomposition of G is the n-tuple $(a_0, \ldots, a_n)$, where a_h denotes the number of bags that have exactly $n - h$ vertices. A tree decomposition of lexicographically minimal fatness is called an *atomic tree decomposition*.

Proposition 7.5. [46, Lemma 3.9]. *Let $(\{X_i\}, T)$ be an atomic tree decomposition of a connected graph $G = (V, E)$. Then, for any two distinct vertices x, y that occur together in some bag X_t, either $(x, y) \in E$ or there exists a neighbor s of t in T such that $\{x, y\} \subseteq V_s \cap V_t$.*

We also need the following proposition, which follows from Lemmas 3.7 and 3.8 in [46].

Proposition 7.6. *Let $(\{X_i\}, T)$ be an atomic tree decomposition of a connected graph G, $e = (s, t) \in E(T)$ be any edge and let T_t be the connected component of $T - e$ rooted at t, and set $X = X_s \cap X_t$. Then there exists a connected component C_t in $G_t \setminus X$ such that $N(C_t) = X$ and $X_t \subseteq C_t \cup X$.*

Finally, we are ready to give a bound on treelength in terms of a graph's treewidth and its longest geodesic cycle. Our proof relies heavily on tools from [46].

Theorem 7.7. *[86]. For any graph $G = (V, E)$, it holds that $tl(G) \leq v(G) \cdot (tw(G)+1)$ where $v(G)$ is the length of the longest geodesic cycle in G.*

Proof. We will prove a stronger statement, namely that *any atomic tree decomposition* of a two-connected graph has treelength at most $v(G) \cdot (tw(G) + 1)$. Let us first show how this proves the lemma for graphs that are not two-connected.

Assume G is not two-connected and $x \in V$ is a cut vertex ($G - x$ has at least two connected components). Let $H_1, \ldots, H_\ell$ be the connected components of $G - x$. If we prove that the graphs $G[H_i \cup \{x\}]$, $1 \leq i \leq \ell$ have tree decompositions $\mathcal{T}_i$ with treelength bounded as in the statement of the theorem, then we can easily construct a tree decomposition for G with the same property: we simply introduce a single new bag $V_x = \{x\}$ and connect it to an arbitrary bag containing x in each of the individual tree decompositions $\mathcal{T}_i$ (because these graphs all contain the vertex x such a bag must exist). Note that the treelength of this decomposition is simply $\max_{1 \leq i \leq \ell} tl(\mathcal{T}_i)$ because the bag V_x we added contains only the vertex x and, thus, cannot increase the treelength. Since we will show the statement for two-connected graphs in the following, we recursively decompose the graph G over cut vertices until the remaining connected components are all two-connected and then construct a tree decomposition of G as described earlier.

We may now assume that G is two-connected. Given an atomic tree decomposition $(\{X_i\}, T)$ of G, we show that for every two vertices x, y that occur in a common bag $X := X_t$, x and y are connected by a path whose length depends only on $|X|$ and $v(G)$. To this end, let $\mathcal{C}_X$ be the collection of geodesic cycles in G that have at least one vertex in X. We first show that if $G[\mathcal{C}_X]$ is connected and $X \subseteq V(\mathcal{C}_X)$, then every pair of vertices in X is connected by a path of length at most $|X| \cdot v(G[\mathcal{C}_X])$.

Consider $x, y \in X$. Start a breadth-first search (bfs) from x that stops as soon as it reaches y. Let $L_1, L_2, \cdots L_p$ be the layers of the bfs-tree where $L_1 = \{x\}$ is the starting layer. We claim that for all L_i with $L_i \cap X \neq \emptyset$, there is a j such that $i < j \leq i + v(G[\mathcal{C}_X])$ and $L_j \cap X \neq \emptyset$. Consider such an L_i, and denote by $X_l \subseteq X$ those vertices of X that are contained in $\bigcup_{k=1}^i L_k$. Denote by $X_r = X \setminus X_l$ those vertices of X that have not been visited until step i. If there exists a geodesic cycle C in $\mathcal{C}_X$ with vertices in both X_l and X_r, we are done – the bfs will have seen all of C in at most $v(G[\mathcal{C}_X])$ steps (and thus found $C \cap X_r$). Otherwise, since $\mathcal{C}_X$ is connected, there exist two geodesic cycles $C_l, C_r \in \mathcal{C}_X$ with $C_l \cap X_l \neq \emptyset$, $C_r \cap X_r \neq \emptyset$ and $C_r \cap C_l \neq \emptyset$. Because the bfs will visit all vertices of $C_r \cup C_l$ in at most $(|C_r| + |C_l|)/2 \leq v(G[\mathcal{C}_X])$ steps, the claim follows. Therefore, the number of layers $p \leq v(G[\mathcal{C}_X]) \cdot |X_t|$, and thus, the distance between x and y is bounded by $v(G[\mathcal{C}_X]) \cdot |X_t|$, as claimed.

Therefore, if we show that for every bag X, the set $\mathcal{C}_X$ of geodesic cycles touching X induces a connected graph $G[V(\mathcal{C}_X)]$, we are done: then, every vertex pair $x, y \in X$ is indeed connected by a path of length at most $v(G[V(\mathcal{C}_X)])|X|$, which (by the definition of treewidth and the fact $\mathcal{C}_X$ is a family of geodesic cycles) is bounded by $v(G)(tw(G)+1)$.

We first prove that for any choice of $X := X_t$ and any pair of vertices $x, y \in X$, x and y lie on some cycle of G. By Proposition 7.5, the vertices x, y are either connected by an edge (in which case we are done: G is two-connected, so every edge lies on some cycle) or there exists some node $s \in N_T(t)$ such that $\{x, y\} \subseteq V_s \cap V_t$. In the latter case, we invoke Proposition 7.6: for $i \in \{s, t\}$ we can find connected components H_i of $G_i \setminus X$

such that $N(H_i) = X$ and $V_i \subseteq H_i \cup X$. Therefore, there exist two x-y-paths: one inside H_s and another in H_t, hence, x and y lie on a cycle.

Because the set of geodesic cycles forms a basis for the cycle space of a graph (see Theorem 3.1 of [54]), it follows that for every $t \in T$, $G[V(\mathcal{C}_{X_t})]$ is connected. The distance between any vertices in X_t is thus bounded by $\nu(\mathcal{C}_{X_t}) \cdot |X_t|$, implying that $tl(G)$ is at most $\nu(G) \cdot (tw(G) + 1)$, as claimed. $\qquad\square$

Finally, we put all the pieces together and show why these bounds are tight.

Proof of Theorem 7.3. This follows directly from combining Theorem 7.7 with the result that hyperbolicity is at most the treelength (Theorem 4 in [40]), and the observation that for any nonnegative integers n and k, the k-subdivision of the $n \times n$ planar grid has treelength $n(k + 1)$, treewidth n, a longest geodesic cycle of length $4(k + 1)$, and hyperbolicity $(n - 1)(k + 1) - 1$. $\qquad\square$

8. DISCUSSION AND CONCLUSION

Clearly, there is a need to develop TD heuristics that are better suited for the properties of realistic informatics graphs. This might involve making more sophisticated choices than greedily minimizing degree or fill, but it might also involve optimizing other parameters such as treelength (which has connections with δ-hyperbolicity) or minimizing the width of bags that are not central (associated with the deep core). In addition, it would be interesting to use TDs to help to combine small local clusters found with other methods, e.g., local spectral methods, into larger overlapping clusters, in order to understand better what might be termed the "local to global" properties of realistic informatics graphs. Because these graphs are not well described by simple low-dimensional structures or simple constant-degree expander-like structures, this coupling is particularly counterintuitive, but it is very important for applications such as the diffusion of information. Finally, given the connections between TDs and graphical models, it would be interesting to understand better the implications of our results for improved graphical modeling and/or for improved inference on realistic network data. We expect that this will be a particularly challenging but promising direction for future work on social (as well as nonsocial) graphs.

REFERENCES

[1] M. M. Abu-Ata. "Tree-Like Structure in Graphs and Embeddability to Trees." PhD thesis, Kent State University, 2014.

[2] M. Abu-Ata and F. F. Dragan. "Metric Tree-Like Structures in Real-Life Networks: An Empirical Study." *Networks* 67:1 (2016), 49–68.

[3] L. A. Adamic and N. Glance. "The Political Blogosphere and the 2004 U.S. Election: Divided They Blog." In *LinkKDD '05: Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43. New York, NY: ACM, 2005.

[4] A. B. Adcock. "Characterizing, Identifying, and Using Tree-Like Structure in Social and Information Networks." PhD thesis, Stanford University, 2014.

[5] A. B. Adcock, B. D. Sullivan, O. R. Hernandez, and M. W. Mahoney. "Evaluating OpenMP Tasking at Scale for the Computation of Graph Hyperbolicity." In *Proc. of the 9th IWOMP*, pp. 71–83. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[6] A. B. Adcock, B. D. Sullivan, and M. W. Mahoney. "Tree-Like Structure in Large Social and Information Networks." In *Proc. of the 2013 IEEE ICDM*, IEEE, 2013.

[7] J. M. Alonso, T. Brady, D. Cooper, V. Ferlini, M. Lustig, M. Mihalik, H. Shapiro, and H. Short. "Notes on Word Hyperbolic Groups." In *Group Theory from a Geometrical Viewpoint, ICTP Trieste Italy*, edited by E. Ghys, A. Haeiger, and A. Verjovski, pp. 3–63. River Edge, NJ: World Scientific, 1991.

[8] P. R. Amestoy, T. A. Davis, and I. S. Duff. "An Approximate Minimum Degree Ordering Algorithm." *SIAM Journal on Matrix Analysis and Applications* 17:4 (1996), 886–905.

[9] P. R. Amestoy, T. A. Davis, and I. S. Duff. "Algorithm 837: AMD, an Approximate Minimum Degree Ordering Algorithm." *ACM Transactions on Mathematical Software (TOMS)* 30:3 (2004), 381–388.

[10] E. Amir. "Efficient Approximation for Triangulation of Minimum Treewidth." In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 7–15. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2001.

[11] E. Amir. "Approximation Algorithms for Treewidth." *Algorithmica* 56:4 (2010), 448–479.

[12] R. Andersen, F. R. K. Chung, and K. Lang. "Local Graph Partitioning Using PageRank Vectors." In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 475–486. IEEE, 2006.

[13] S. Arnborg and A. Proskurowski. "Linear Time Algorithms for NP-Hard Problems Restricted to Partial K-Trees." *Discrete Applied Mathematics* 23:1 (1989), 11–24.

[14] V. Batagelj and A. Mrvar. "Pajek—Analysis and Visualization of Large Networks." In *Proceedings of Graph Drawing*, pp. 477–478. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[15] V. Batagelj and M. Zaversnik. "An $O(m)$ algorithm for cores decomposition of networks." Technical Report. 2003. Preprint: arXiv:cs.DS/0310049.

[16] V. Batagelj and M. Zaversnik. "Generalized cores." Technical Report. 2002. Preprint: arXiv:cs.DS/0202039.

[17] V. Batagelj and M. Zaversnik. "Fast Algorithms for Determining (Generalized) Core Groups in Social Networks." *Advances in Data Analysis and Classification* 5:2 (2011), 129–145.

[18] A. Becker and D. Geiger. "A Sufficiently Fast Algorithm for Finding Close to Optimal Clique Trees." *Artificial Intelligence* 125:1–2 (2001), 3–17.

[19] P. Bellenbaum and R. Diestel. "Two Short Proofs Concerning Tree-Decompositions." *Combinatorics, Probability, and Computing* 11 (2002), 541–547.

[20] M. W. Bern, E. L. Lawler, and A. L. Wong. "Linear-Time Computation of Optimal Subgraphs of Decomposable Graphs." *Journal of Algorithms* 8:2 (1987), 216–235.

[21] A. Berry, J. R. S. Blair, and P. Heggernes. "Maximum Cardinality Search for Computing Minimal Triangulations." In *Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 1–12. London, UK: Springer Verlag, 2001.

[22] A. Berry, J. R. S. Blair, P. Heggernes, and B. W. Peyton. "Maximum Cardinality Search for Computing Minimal Triangulations of Graphs." *Algorithmica* 39:4 (2004), 287–298.

[23] A. Berry, P. Heggernes, and G. Simonet. "The Minimum Degree Heuristic and the Minimal Triangulation Process." In *Graph-Theoretic Concepts in Computer Science*, edited by H. L. Bodlaender, pp. 58–70, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[24] J. R. S. Blair and B. Peyton. "An Introduction to Chordal Graphs and Clique Trees." In *Graph Theory and Sparse Matrix Computation*, edited by A. George, J. R. Gilbert, and J. W. H. Liu, pp. 1–29, The IMA Volumes in Mathematics and its Applications, 56. New York, NY: Springer New York, 1993.

[25] H. L. Bodlaender. "A Tourist Guide Through Treewidth." *Acta Cybernetica* 11 (1993), 1–23.

[26] H. L. Bodlaender. "A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth." *SIAM Journal on Computing* 25:6 (1996), 1305–1317.

[27] H. L. Bodlaender. "Discovering Treewidth." In *Proceedings of the 31st International Conference on Theory and Practice of Computer Science*, pp. 1–16. Berlin, Heidelberg: Springer-Verlag, 2005.

[28] H. L. Bodlaender. "Treewidth: Characterizations, Applications, and Computations." In *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 1–14. Berlin, Heidelberg: Springer-Verlag, 2006.

[29] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks. "Approximating Treewidth, Pathwidth, and Minimum Elimination Tree Height." *Journal of Algorithms* 18 (1995), 238–255.

[30] H. L. Bodlaender and A. M. C. A. Koster. "Combinatorial Optimization on Graphs of Bounded Treewidth." *The Computer Journal* 51:3 (2007), 255–269.

[31] H. L. Bodlaender and A. M. C. A. Koster. "Treewidth Computations I. Upper Bounds." *Inf. Comput.* 208:3 (2010), 259–275.

[32] H. L. Bodlaender and R. H. Möhring. "The Pathwidth and Treewidth of Cographs." *SIAM Journal on Discrete Mathematics* 6:2 (1993), 181–188.

[33] B. Bollobás. *Random Graphs*. London: Academic Press, 1985.

[34] V. Bouchitté, D. Kratsch, H. Müller, and I. Todinca. "On Treewidth Approximations." *Discrete Appl. Math.* 136:(2-3) (2004), 183–196.

[35] G. Brinkmann, J. H. Koolen, and V. Moulton. "On the Hyperbolicity of Chordal Graphs." *Annals of Combinatorics* 5:1 (2001), 61–69.

[36] C. Chekuri and J. Chuzhoy. "Polynomial Bounds for the Grid-Minor Theorem." In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 60–69. 2014.

[37] H. Chen. "Quantified Constraint Satisfaction and Bounded Treewidth." In *Proceedings of the 16th European Conference on Artificial Intelligence*, R. López de Mántaras and L. Saitta (eds.), pp. 161–165. Amsterdam: IOS Press, 2004.

[38] W. Chen, W. Fang, G. Hu, and M. W. Mahoney. "On the Hyperbolicity of Small-World and Tree-Like Random Graphs." *Internet Mathematics* 9:4 (2013), 434–491. Also available at: arXiv:1201.1717.

[39] J. Cheng, Y. Ke, S. Chu, and M. T. Ozsu. "Efficient Core Decomposition in Massive Networks." In *Proceedings of the 27th IEEE International Conference on Data Engineering*, pp. 51–62. Washington, DC: IEEE Computer Society, 2011.

[40] V. Chepoi, F. Dragan, B. Estellon, M. Habib, and Y. Vaxès. "Diameters, Centers, and Approximating Trees of δ-Hyperbolic Geodesic Spaces and Graphs." In *Proceedings of the 24th Annual Symposium on Computational Geometry*, pp. 59–68. New York, NY: ACM, 2008.

[41] F. R. K. Chung and L. Lu. *Complex Graphs and Networks*, CBMS Regional Conference Series in Mathematics 107. Boston, MA: American Mathematical Society, 2006.

[42] P. Colomer-de Simon, A. Serrano, M. G. Beiro, J. Ignacio Alvarez-Hamelin, and M. Boguna. "Deciphering the Global Organization of Clustering in Real Complex Networks." *Scientific Reports* 3 (2013), 2517.

[43] B. Courcelle and M. Mosbah. "Monadic Second-Order Evaluations on Tree-Decomposable Graphs." *Theoretical Computer Science* 109:1–2 (1993), 49–82.

[44] T. A. Davis and Y. Hu. "The University of Florida Sparse Matrix Collection." *ACM Transactions on Mathematical Software (TOMS)* 38:1 (2011), 1:1–1:25.

[45] F. de Montgolfier, M. Soto, and L. Viennot. "Treewidth and Hyperbolicity of the Internet." In *Proceedings of the 10th IEEE International Symposium on Network Computing and Applications (NCA)*, pp. 25–32. IEEE, 2011.

[46] R. Diestel and M. Müller. "Connected Tree-Width." Technical Report. Preprint: arXiv:1211.7353. 2012.

[47] Y. Dourisboure and C. Gavoille. "Tree-Decompositions with Bags of Small Diameter." *Discrete Mathematics* 307:16 (2007), 2008–2029.

[48] F. F. Dragan. "Tree-Like Structures in Graphs: A Metric Point of View." In *Proceedings of the 39th International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 1–4. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[49] F. F. Dragan and I. Lomonosov. "On Compact and Efficient Routing in Certain Graph Classes. *Discrete Applied Mathematics* 155:11 (2007), 1458–1470.

[50] P. Erdős and A. Rényi. "On the Evolution of Random Graphs." *Publ. Math. Inst. Hungar. Acad. Sci.* 5 (1960), 17–61.

[51] E. R. Gansner and S. C. North. "An Open Graph Visualization System and Its Applications to Software Engineering." *Software–Practice and Experience* 30:11 (2000), 1203–1233.

[52] Y. Gao. "Treewidth of Erdős-Rényi Random Graphs, Random Intersection Graphs, and Scale-Free Random Graphs." *Discrete Applied Mathematics* 160:4–5 (2012), 566–578.

[53] F. Gavril. "The Intersection Graphs of Subtrees in Trees are Exactly the Chordal Graphs." *Journal of Combinatorial Theory, Series B* 16:1 (1974), 47–56.

[54] A. Georgakopoulos and P. Sprussel. "Geodesic Topological Cycles in Locally Finite Graphs." *The Electronic Journal of Combinatorics* 16:1 (2009), R144.

[55] J. A. George. "Nested Dissection of a Regular Finite Element Mesh." *SIAM Journal of Numerical Analysis* 10 (1973), 345–363.

[56] J. R. Gilbert and R. E. Tarjan. "The Analysis of a Nested Dissection Algorithm." *Numerische Mathematik* 50:4 (1986), 377–404.

[57] C. Groër, B. D. Sullivan, and D. Weerapurage. "INDDGO: Integrated Network Decomposition & Dynamic Programming for Graph Optimization." Technical Report ORNL/TM-2012/176, Oak Ridge National Laboratory, 2012.

[58] M. Grohe and D. Marx. "On Tree Width, Bramble Size, and Expansion." *Journal of Combinatorial Theory Series B* 99:1 (2009), 218–228.

[59] M. Gromov. "Hyperbolic Groups." In *Essays in Group Theory*, edited by S. M. Gersten, pp. 75–263, Math. Sci. Res. Inst. Publ. 8. New York, NY: Springer, 1987.

[60] J. Healy, J. Janssen, E. Milios, and W. Aiello. "Characterization of Graphs Using Degree Cores." In *WAW '08: Proceedings of the 6th Workshop on Algorithms and Models for the Web-Graph*, pp. 137–148. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[61] P. Heggernes. "Minimal Triangulations of Graphs: A Survey." *Discrete Mathematics* 306:3 (2006), 297–317.

[62] I. V. Hicks, A. M. C. A. Koster, and E. Kolotoğlu. "Branch and Tree Decomposition Techniques for Discrete Optimization." In *INFORMS Tutorials in Operations Research: Emerging Theory, Methods, and Applications*, pp. 1–29. 2005.

[63] J. Ignacio Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. "Large Scale Networks Fingerprinting and Visualization Using the K-Core Decomposition." In *Annual Advances in*

Neural Information Processing Systems 18: Proceedings of the 2005 Conference, pp. 41–50. MIT Press, 2006.

[64] J. Ignacio Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. "K-Core Decomposition of Internet Graphs: Hierarchies, Self-Similarity and Measurement Biases." *Networks and Heterogeneous Media* 3:2 (2008), 371–393.

[65] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney. "Think Locally, act locally: Detection of Small, Medium-Sized, and Large Communities in Large Networks." *Physical Review E* 91 (2015), 012821.

[66] E. A. Jonckheere, P. Lohsoonthorn, and F. Ariaei. "Scaled Gromov Four-Point Condition for Network Graph Curvature Computation." *Internet Mathematics* 7:3 (2011), 137–177.

[67] E. A. Jonckheere, P. Lohsoonthorn, and F. Bonahon. "Scaled Gromov Hyperbolic Graphs." *Journal of Graph Theory* 57:2 (2008), 157–180.

[68] D. Karger and N. Srebro. "Learning Markov Networks: Maximum Bounded Tree-Width Graphs." In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pp. 392–401. 2001.

[69] G. Karypis and V. Kumar. "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs." *SIAM Journal on Scientific Computing* 20 (1998), 359–392.

[70] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse. "Identification of Influential Spreaders in Complex Networks." *Nature Physics* 6:11 (2010), 888–893.

[71] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press, 2009.

[72] A. Kosowski, B. Li, N. Nisse, and K. Suchan. "K-Chordal Graphs: From Cops and Robber to Compact Routing via Treewidth." In *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming*, pp. 610–622. Berlin Heidelberg: Springer Berlin Heidelberg, 2012.

[73] A. M. C. A. Koster, H. L. Bodlaender, and S. P. M. van Hoesel. "Treewidth: Computational Experiments." *Electronic Notes in Discrete Mathematics* 8 (2001), 54–57.

[74] A. M. C. A. Koster, S. P. M. van Hoesel, and A. W. J. Kolen. "Solving Partial Constraint Satisfaction Problems with Tree Decomposition." *Networks* 40:3 (2002), 170–180.

[75] J. Lagergren. "Efficient Parallel Algorithms for Graphs of Bounded Tree-Width." *Journal of Algorithms* 20:1 (1996), 20–44.

[76] S. L. Lauritzen and D. J. Spiegelhalter. "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems (with Discussion)." *Journal of the Royal Statistical Society series B* 50 (1988), 157–224.

[77] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. "Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters." *Internet Mathematics* 6:1 (2009), 29–123. Also available at: arXiv:0810.1355.

[78] C. Liu, Y. Song, B. Yan, Y. Xu, and L. Cai. "Fast de Novo Peptide Sequencing and Spectral Alignment via Tree Decomposition." In *Pacific Symposium on Biocomputing*, pp. 255–266. World Scientific, 2006.

[79] D. Lokshtanov. "On the Complexity of Computing Treelength." In *Proceedings of the 32nd International Conference on Mathematical Foundations of Computer Science*, pp. 276–287. Amsterdam, The Netherlands: Elsevier Science Publishers B. V., 2007.

[80] T. Maehara, T. Akiba, Y. Iwata, and K. Kawarabayashi. "Computing Personalized PageRank Quickly by Exploiting Graph Structures." *Proceedings of the VLDB Endowment* 7 (2014), 1023–1034.

[81] T. Malisiewicz. "Open Source Code: Graphviz MATLAB Magic." Available online (`https://github.com/quantombone/graphviz_matlab_magic`), May 2010.

[82] H. M. Markowitz. "The Elimination Form of the Inverse and Its Application to Linear Programming." *Management Science* 3:3 (1957), 255–269.

[83] A. G. Percus, G. Istrate, B. Goncalves, R. Z. Sumi, and S. Boettcher. "The Peculiar Phase Structure of Random Graph Bisection." *Journal of Mathematical Physics* 49:12 (2008), 125219.

[84] V. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, A. Gupta, and A. Akella. "On the Treeness of Internet Latency and Bandwidth." In *Proceedings of the 2009 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 61–72. 2009.

[85] B. A. Reed. "Finding Approximate Separators and Computing Tree Width Quickly." In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pp. 221–228. New York, NY: ACM, 1992.

[86] F. Reidl and B. Sullivan. Personal email communication, November 2014.

[87] N. Robertson and P. D. Seymour. "Graph Minors. II. Algorithmic Aspects of Tree-Width." *Journal of Algorithms* 7:3 (1986), 309–322.

[88] H. Röhrig. "Tree Decomposition: A Feasibility Study." Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 1998.

[89] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha. "Core-Periphery Structure in Networks." *SIAM Journal on Applied Mathematics* 74:1 (2014), 167–190.

[90] D. J. Rose and R. E. Tarjan. "Algorithmic Aspects of Vertex Elimination." In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing*, pp. 245–254. New York, NY: ACM, 1975.

[91] D. Rose, R. Tarjan, and G. Lueker. "Algorithmic Aspects of Vertex Elimination on Graphs." *SIAM Journal on Computing* 5 (1976), 266–283.

[92] S. B. Seidman. "Network Structure and Minimum Degree." *Social Networks* 5:3 (1983), 269–287.

[93] P. D. Seymour and R. Thomas. "Call Routing and the Ratcatcher." *Combinatorica* 14:2 (1994), 217–241.

[94] Y. Shavitt and T. Tankel. "Hyperbolic Embedding of Internet Graph for Distance Estimation and Overlay Construction." *IEEE/ACM Transactions on Networking* 16:1 (2008), 25–36.

[95] K. Shoikhet and D. Geiger. "A Practical Algorithm for Finding Optimal Triangulations." In *Proceedings of AAAI/IAAI*, pp. 185–190. New York, NY: AAAI, 1997.

[96] B. D. Sullivan et al. "Integrated Network Decompositions and Dynamic Programming for Graph Optimization (INDDGO)." Available online (http://github.com/bdsullivan/inddgo) 2012, 2013.

[97] Supporting website. http://snap.stanford.edu/data/index.html.

[98] R. E. Tarjan and M. Yannakakis. "Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs." *SIAM Journal on Computing* 13 (1984), 566–579.

[99] R. E. Tarjan and M. Yannakakis. "Addendum: Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs." *SIAM Journal on Computing* 14:1 (1985), 254–255.

[100] A. L. Traud, P. J. Mucha, and M. A. Porter. "Social Structure of Facebook Networks." *Physica A* 391 (2012), 4165–4180.

[101] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. "Structural Diversity in Social Contagion." *Proceedings of the National Academy of Sciences* 109:16 (2012), 5962–5966.

[102] K. Verbeek and S. Suri. "Metric Embedding, Hyperbolic Space, and Social Networks." In *Proceedings of the 30th Annual Symposium on Computational Geometry*, pp. 501–510. New York, NY: ACM, 2014.

[103] C. Wang, T. Liu, P. Cui, and K. Xu. "A Note on Treewidth in Random Graphs." In *Proceeding of the 5th International Conference on Combinatorial Optimization and Applications*, pp. 491–499. Berlin, Heidelberg: Springer-Verlag, 2011.

[104] D. J. Watts and S. H. Strogatz. "Collective Dynamics of Small-World Networks." *Nature* 393 (1998), 440–442.

[105] Y. Wu and C. Zhang. "Hyperbolicity and Chordality of a Graph." *The Electronic Journal of Combinatorics* 18:1 (2011), P43.

[106] J. Zhao, D. Che, and L. Cai. "Comparative Pathway Annotation with Protein-DNA Interaction and Operon Information via Graph Tree Decomposition." In *Pacific Symposium on Biocomputing*, pp. 496–507. World Scientific, 2007.

[107] J. Zhao, R. L. Malmberg, and L. Cai. "Rapid *ab initio* RNA Folding Including Pseudoknots via Graph Tree Decomposition." In *Proceedings of the 6th International Workshop on Algorithms in Bioinformatics*, pp. 262–273. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.