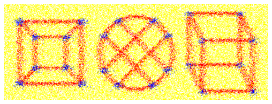

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 19, no. 2, pp. 595–618 (2015)
DOI: 10.7155/jgaa.00370

Untangling the Hairballs of Multi-Centered, Small-World Online Social Media Networks

Arlind Nocaj Mark Ortmann Ulrik Brandes

Department of Computer & Information Science
University of Konstanz

Abstract

Small-world graphs have characteristically low average distance and thus cause force-directed methods to generate drawings that look like hairballs. This is by design as the inherent objective of these methods is a globally uniform edge length or, more generally, accurate distance representation. The problem arises, for instance, with graphs of high density or high conductance, or in the presence of high-degree vertices, all of which tend to pull vertices together and thus result in clutter overspreading variation in local density.

We here propose a method specifically for a class of small-world graphs that are typical for online social networks. The method is based on a spanning subgraph that is sparse but connected and consists of strong ties holding together communities. To identify these ties we propose a novel criterion for structural embeddedness. It is based on a weighted accumulation of triangles in quadrangles and can be determined efficiently. An evaluation on empirical and generated networks indicates that our approach improves upon previous methods using other edge indices. Although primarily designed to achieve more informative drawings, our spanning subgraph may also serve as a sparsifier that trims a small-world graph prior to the application of a clustering algorithm.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
November 2014	September 2015	September 2015	September 2015	September 2015

Published:
November 2015

Article type:
Regular paper

Communicated by:
C. Duncan and A. Symvonis

This research was supported by DFG under grants GRK 1042, Br 2158/6-1, and Br 2158/11-1.
The proposed method is available in visone.

E-mail addresses: Arlind.Nocaj@uni-konstanz.de (Arlind Nocaj) Mark.Ortmann@uni-konstanz.de (Mark Ortmann) Ulrik.Brandes@uni-konstanz.de (Ulrik Brandes)

1 Introduction

Online social networks such as Facebook friendship graphs are an amalgamation of a variety of social relations. The presence of a friendship tie might be due to shared interests, spatial proximity, kinship, or professional relations to name but a few. When such a multitude of relations is conflated in the same network, any two nodes are likely to be connected via at most a few links – thus leading to a *small-world effect* [35]. Visualizations of these graphs using standard layout methods such as force-directed placement produce drawings in which variation in local structure is hidden in a densely-looking, overlap-ridden *hairball*.

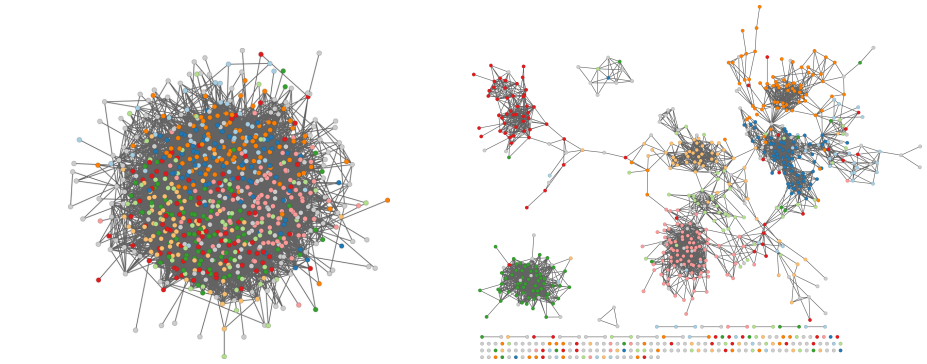
Hairball drawings, as for example shown in Fig. 1(a), however, are not only the result of small-world graphs, but of any graph which exhibits a low variation in pairwise shortest path distances. In the following we refer to graphs with this characteristic as *hairball graphs*.

Various approaches to reduce the clutter in drawings of small worlds and other hairball graphs have been proposed [21], most notably *edge bundling* [17, 30], *edge lensing* [20], modified *layout algorithms* [3] or *representations* [1, 11, 27, 43], and *graph simplification* [2, 29, 32, 34, 37, 42, 44]. The idea of graph simplification is to identify a subset of edges such that only the resulting graph, the so-called backbone, needs to be laid out. We adopt this approach and propose a new method to trim hairball graphs.

Problem formulations in graph simplification include the preservation of properties such as cuts [2], spectra [37, 38], connectivity [44], collapsing substructures into supernodes [32], and emphasizing certain connections [29, 34]. As graph invariants such as cuts are more easily affected by noise in empirical networks, we opt for locally defined graph simplification criteria. As a simplification criterion we use the concept of structural embeddedness in social networks. While the *strength* of a social tie is an inherent characteristic of an individual relationship [14], *embeddedness* refers to the density of the graph around that edge [8, 15].

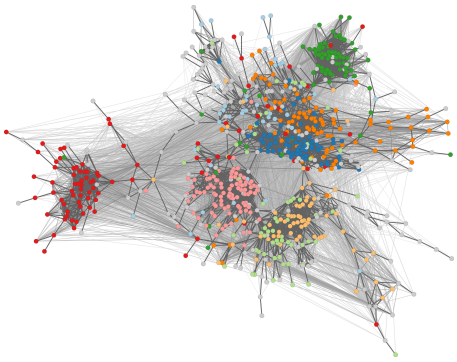
In line with sociological ideas of Simmel [36], Satuliri et al. [34] determine the embeddedness of an edge as the fraction of common neighbors. The Simmelian backbone of Nick et al. [29] introduces an additional local adaptation step that starts from an initial weight – a strength or embeddedness criterion such as the number of triangles an edge is contained in – and then reweights each edge by comparing the ranked neighborhoods of its two vertices. In both methods, the backbone is obtained by finally removing all edges with weights below a specified nodal or network-wide threshold.

These filtering techniques are related to graph partitioning techniques based on edge weights [28]. Since we want to use them for graph drawing, a major difference is that we actually want to maintain connectedness. Otherwise, the layout algorithm is oblivious to edges of the original graph connecting vertices in different components of the backbone as, for example, in Fig. 1(b). When connected components happen to be placed far apart, these edges will run across the drawing and produce even worse clutter.

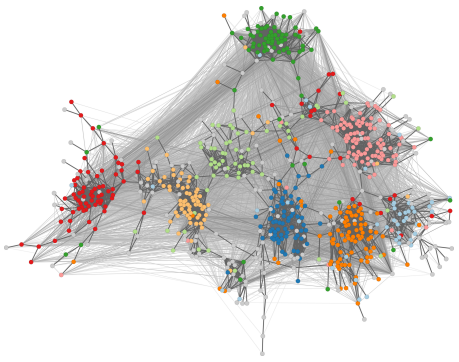


(a) hairball drawing, original network

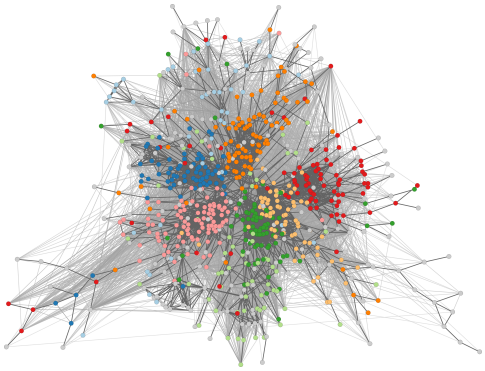
(b) triadic Simmelian backbone [29]



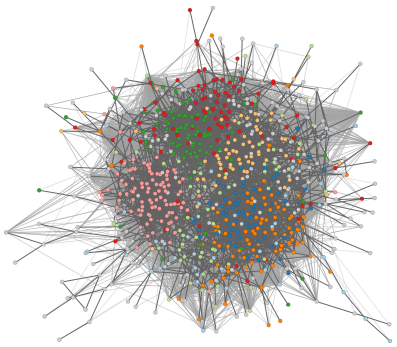
(c) triadic Simmelian backbone with UMST



(d) quadrilateral Simmelian backbone with UMST (our approach)



(e) Jaccard [34] with UMST



(f) Density [1] with UMST

Figure 1: Facebook friendships at California Institute of Technology (Caltech36). Vertex color corresponds to dormitory (gray for missing values), but has not been utilized by the layout algorithm. The layout in (a) is based on the entire graph, whereas (b)-(f) use edge embeddedness, which spreads the graph while keeping cohesive groups together. Embeddedness mapped to edge color; backbone edges dark gray.

We present an efficient preprocessing technique that allows to draw a certain class of small-world social networks with standard layout algorithms that would produce hairball layouts otherwise. Our main contributions are:

- a novel method to identify deeply embedded ties,
- the use of the union of all maximum spanning trees as a sparsifier that maintains connectedness and avoids subtree-ordering ambivalence, and
- an evaluation on observed and generated networks.

We outline our overall method for drawing hairball graphs in the next section and describe our edge embeddedness metric in Sect. 3. Different metrics are evaluated in Sect. 4 and we conclude in Sect. 5.

2 Drawing Algorithm

The main challenges in drawing hairball graphs are their high density, low diameter and noisy group structure. Therefore, our goal is to find a backbone of the graph that retains deeply embedded edges and thus can be used to draw the original graph, e.g., by a force-directed method [22] to reveal the actual variation in cohesiveness.

Since most drawing methods cannot put vertices of different graph components into a meaningful spatial relation, cf. Fig. 1(b), we need to maintain the graph connectivity to retain the global context.

This leads to the following requirements on our backbone:

(i) Edges should be favored based on their structural embeddedness only.

(ii) Connectedness has to be maintained.

Two common approaches to simplify a graph $G = (V, E, w)$ with vertex set V, edge set E, and edge weight $w : E \to \mathbb{R}_{\geq 0}$, are *sampling* [2, 37] and *thresholding* [1, 29, 34]. Note that we assume that w reflects the embeddedness of an edge and a higher value corresponds to stronger embeddedness. Although sampling can be used for sparsification purposes, the random selection of edges violates both of our requirements. In contrast, thresholding guarantees that edges are favored by their weights and consequently their structural properties, as it retains only the top k percent of edges with respect to w. Nevertheless, neither nodal nor network wide thresholding can ensure that the backbone stays connected.

Sparse connected subgraphs of edges not likely to be between cohesive groups have been proposed, e.g., by van Ham and Wattenberg [42] (planar graphs) and Tumminello et al. [41] (graph of bounded genus). A minimally connected subgraph of edges with high weights is a *maximum spanning tree* (MST), and Mantegna [24] proposed these as a backbone. Trees, however, have severe drawbacks: firstly, they do not maintain any local variation in density and, secondly,

Algorithm 1: Hairball Drawing Algorithm

Input: Undirected Graph $G = (V, E)$ and sparsification ratio $s \in [0, 1]$.
Output: Vertex positions $P \in \mathbb{R}^{|V| \times 2}$

$w \leftarrow$ embeddedness weights of edges
sort edges by non-increasing weight
$E_{\text{union}} \leftarrow$ UMST with respect to w
$E_{\text{threshold}} \leftarrow \{e \in E : w(e) \geq w(e_{\lceil (1-s)|E| \rceil})\}$
$P \leftarrow$ layout determined from spanning subgraph $(V, E_{\text{union}} \cup E_{\text{threshold}})$

Algorithm 2: UMST: Union of all Maximum Spanning Trees

Input: Undirected Graph $G = (V, E)$ and edge weights $w : E \to \mathbb{R}_{\geq 0}$.
Data: Union-Find datastructure
Output: Edges belonging to any MST

```
E_union ← ∅
partition edges by weight into buckets B_1, ..., B_k
sort buckets by decreasing weight
for i ← 1 to k do
    M ← ∅
    foreach e = (u, v) ∈ B_i do
        if find(u) ≠ find(v) then M ← M ∪ {e}
    foreach e = (u, v) ∈ M do union(u, v)
    E_union ← E_union ∪ M
```

they introduce a subtree ordering ambiguity. While the first also means that arbitrary choices must be made when edges have equal embeddedness, the second creates a degree of freedom that is almost as bad as disconnected components.

We combine thresholding (to maintain local variation) with the union of all maximum spanning trees (UMST; to maintain connectedness). The UMST does not only solve the problem of tie breaks but also reduces the ordering problem by resulting in higher connectivity (Fig. 1(b)-(d)).

The complete algorithm to compute the layout of a small-world graph is presented in Alg. 1. Note that the UMST only contributes the (heaviest) edges necessary to connect the components that result from the thresholding process.

Kruskal's algorithm [9] for minimum spanning trees is easily adapted to determine the union of all maximum spanning trees. Since every edge of maximum weight that has not been processed yet could be chosen next, we batch-process them before components are merged; cf. Algorithm 2. Given that the edges are sorted by their weights, the runtime of Alg. 2 is in $\mathcal{O}(m\tau(m, n))$ with τ being the functional inverse of the Ackermann function [9], which is *practically* a small constant.

The final layout emphasizes variation in local density by considering only deeply embedded edges as expressed by the weights introduced in the next section.

3 Edge Embeddedness by Accumulating Triadic Effects

Real world networks are often aggregates of different relations, which can hamper the detection of subgroups or clusters. Our goal is to determine deeply embedded edges, which are likely to be inside of cohesive groups, so that we can use them to emphasize the inherent structure. The assumption here is that an edge linking a vertex to another vertex in the same subgroup of a network is more embedded than an edge to a vertex outside of that group.

Satuliri et al. [34] propose to capture the embeddedness of an edge $e = (u, v)$ by the Jaccard coefficient over u's and v's neighborhood. Nick et al. [29] suggest a more general framework, consisting of the following main steps:

1. For each edge, determine its weight. *(weighting)*
2. For each vertex, rank all its neighbors acc. to the edge weight. *(sorting)*
3. For each edge, adapt its weight based on the ranking. *(reweighting)*

The approach of Satuliri et al. can be seen as using a uniform edge weight for step one and the Jaccard coefficient for the reweighting in step three. Contrary to this, Nick et al. use the number of triangles an edge is embedded in (its *Simmelianness* [10]) for step one and the maximum prefix Jaccard coefficient for step three. The latter chooses k such that the Jaccard coefficient of the first top k ranked neighbors of u and v is maximized. The effect of this ranking measure is that the highly ranked neighbors have more importance attached, since fewer common vertices are needed to get a high coefficient.

A more intuitive interpretation of this framework is that for an edge $e = (u, v)$ the edge weight allows us to determine the most important neighbors of u and v. If these most important neighbors are the same, e is deeply embedded; otherwise e is connecting two vertices, which are likely to be in different groups. We follow the main idea, but propose a different edge weight than the number of triangles.

Consider the setting in Fig. 2. Clearly, edge e is strongly embedded. Compared to all other edges it closes many triangles resulting in an increase of the *group performance* [6] by introducing mediator effects. Similar to this, an edge (s, t) connecting two triangles at e introduces additional mediator effects on the triangles, which in turn increases the importance of e. We call these edges *mediator edges* on e.

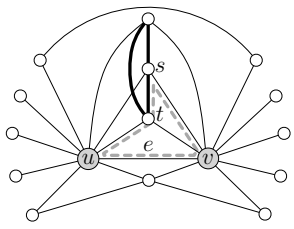


Figure 2: Triangles at edge e [29, 34] do not capture *mediator edges* (bold), while quadrangles do.

Counting the number of triangles at e does not capture the importance of mediator edges. But since each mediator edge creates two quadrangles at e, cf. dashed-contour in Fig. 2, we can use the number of quadrangles containing e to capture this mediator effect. While there can be additional

quadrangles at e, they will be counted only once from e's perspective, which makes their influence rather low. Furthermore, counting the two different types of quadrangles at e would be too time consuming and therefore we will not distinguish between them.

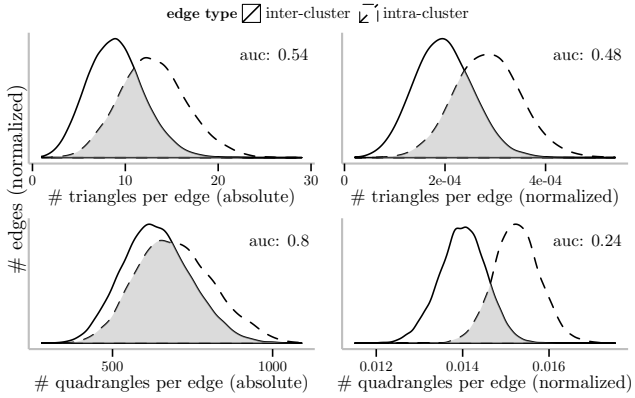


Figure 3: Density distribution for the number of triangles and quadrangles per edge for a network from a planted partition model (500 nodes and 9 clusters). Gray *area under the curve* (auc) corresponds to the error made by distinguishing between intra-/inter-cluster edges using the corresponding feature. While normalization reduces this error in general, the normalized number of quadrangles discriminates better between the two edge types.

Using the absolute number of quadrangles poses difficulties, when the network contains subgroups of different densities. Hence, we normalize this absolute value by putting it into relation to all edges at vertex u and v.

Figure 3 shows the distribution of the number of triangles and quadrangles per edge for a synthetic network with 500 vertices and 9 denser subgroups, generated using the planted partition model (Sect. 4). While the triangle feature discriminates better between intra-/inter-cluster edges using the absolute value, the quadrangle feature clearly dominates when normalized, which becomes obvious by comparing the gray area under the curve.

Let $q(u, v)$ be the number of quadrangles containing edge $(u, v) \in E$. We define the *quadrilateral* edge embeddedness as

$$Q(u, v) = \frac{q(u, v)}{\sqrt{q(u) \cdot q(v)}},$$

where $q(v) = \sum_{w \in N(v)} q(v, w)$, for $v \in V$, and $N(v)$ the neighborhood of v. We use the geometric mean over the arithmetic mean, since it takes the dependency of two variables into stronger consideration [16].

Note that edge-metrics using quadrangles have already been proposed by Auber et al. [1] and Radicchi et al. [33], but are different from our method as they focus on density. For a comparison of different edge metrics we refer the reader to Melançon and Sallaberry [26].

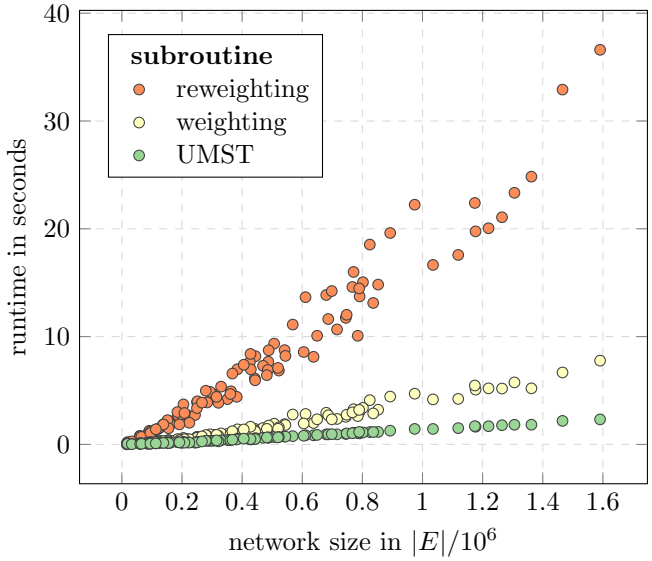


Figure 4: Practical runtimes of quadrilateral Simmelian backbone (with its subroutines) for all Facebook100 networks show scalability of backbone extraction. Edge reweighting is clearly the bottleneck. Using a suitable PDF viewer, a click on the data points reveals network information.

Computation and Time Complexity

We divide the overall backbone extraction into three main steps: edge weighting, edge reweighting, and UMST; For the practical runtime analysis, the sorting is considered as a part of the reweighting step. The respective runtimes for the Facebook100 networks are shown in Fig. 4.

The quadrangles of a graph G can be counted in $\mathcal{O}(m\alpha(G))$ [7], where m is the number of edges and $\alpha(G)$, the *arboricity* of G, is the minimum number of edge-disjoint forests necessary to cover all edges of G. While the arboricity can be as large as $\sqrt{m}$, it is bounded from above by the h-index of a graph which in turn is found to be very small in social networks [12]. An even stronger bound for the arboricity is given by the degeneracy, which is the smallest k such that every subgraph has a vertex of degree at most k. Figure 5 shows that the arboricity is very small, even for large networks of the Facebook100 dataset.

Together with the normalization, the computation of the edge weights takes $\mathcal{O}(m\alpha(G))$ time. Since the counting algorithm of Chiba and Nishizeki [7] for quadrangles and triangles is essentially the same, we refer the reader to [31] for an experimental evaluation on triangle listing algorithms.

Neighbors can be ranked in $\mathcal{O}(m \log \triangle(G))$ time and reweighting can be done in $\mathcal{O}(m\triangle(G))$, where $\triangle(G)$ is the maximum vertex degree, resulting in an overall runtime of $\mathcal{O}(m\triangle(G))$ for the edge reweighting step.

The overall backbone computation (with UMST) took 0.14s on a network

with 762 vertices and 16k edges (Caltech36) and 1.23s on a network with 4087 vertices and 180k edges (Rice31) with our Java 7 implementation and an Intel Core i7-2600K CPU@3.40GHz. Unsurprisingely, the edge reweighting step is also practically the bottleneck, as Fig. 4 reveals. This is due to its $\Delta(G)$ dependency, which, as indicated by Fig. 6 for the Facebook100 networks, cannot be expected to be a small constant.

Nevertheless, the approach scales to large networks and we turn to the evaluation of its effectiveness in the next section.

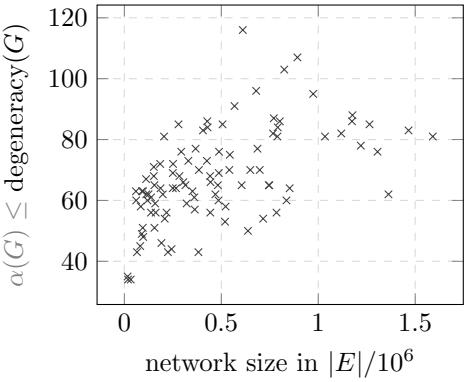


Figure 5: Degeneracy for all Facebook100 networks gives an upper bound for the arboricity $\alpha(G)$ and thus for the asymptotic runtime $O(m\alpha(G))$ of the quadrangle listing algorithm.

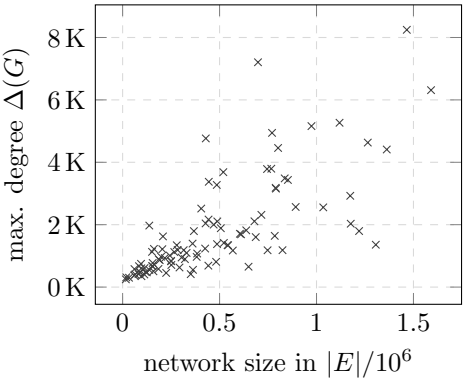


Figure 6: Maximum degree $\Delta(G)$ for all Facebook100 networks. $\Delta(G)$ cannot be expected to be a small constant.

4 Evaluating Methods for Edge Embeddedness

In this section we introduce the datasets and graph models, from which we generate artificial small-world graphs. Then we explain our output quality indicators and the different edge embeddedness methods. For each graph and edge embeddedness method, we iteratively increase the sparsification ratio by 10% and compute the corresponding backbone. Layouts are computed using stress majorization [13] initialized by PivotMDS [4] as suggested in [5]. Note that for larger graphs, we recommend the usage of more scalable force-directed layout methods [18, 19].

4.1 Dataset and Models

As real world samples, we use the Facebook100 dataset [40], which contains social relations of 100 higher educational institutes in the US. The network size varies from 762 to 41K vertices and from 16K to 1.6M edges. The dataset is directly from Facebook, not sampled, and thus very complete in terms of capturing the social relations according to a widely used service at that time. Additional attributes obtained from the Facebook profiles are gender, expected year of graduation, dormitory, etc. Due to incomplete profiles, a number of attribute values are missing. We will use the dormitory attribute for our evaluation, because it has been argued to be important for the creation of social relations in many of the networks [40].

In spite of a strong empirical association with homophilous attribute values, no ground-truth group structure is available for Facebook networks. Therefore, we generated artificial networks that represents the idealized version of multi-core networks, considered in this application, using the *planted partition model* [25].

Additionally, we consider single-centered core-periphery networks; a different type of small-world graphs. The low variation in local density, compared to the multi-core networks, and rather consistent increase of density towards the center usually does not allow for identification of other sub groups than the core or periphery. We used artificial core-periphery networks based on *threshold graphs* [23], as well as the world trade network [39] as a real world example.

Planted Partition Model: A simple model generating random graphs with cohesive groups that are connected into a small world is the *planted partition model* (PPM) [25]. Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a partition of V for a graph $G = (V, E)$. Then $\mathcal{C}$ is called a clustering of G with class $c(v) \in \mathcal{C}$ for a vertex $v \in V$. The probability of an edge (u, v) is p_{in} if $c(u) = c(v)$ and p_{out} if $c(u) \neq c(v)$.

We generated 50 graphs from a PPM with 500 vertices, $k = 9$, $p_{in} = 0.3$, and $p_{out} = 0.01$. On top of that, we ran a random noise model with $p_{in} = p_{out} = 0.1$ to obfuscate the underlying group structure. The resulting graphs are very dense, have a low diameter, and are real hairballs without any visible structure

when laid out using force-directed methods. The presented results of our model are averaged over these 50 samples.

Threshold Graphs: A threshold graph $G = (V, E)$ can be defined by assigning non-negative real weights x_i to each vertex $i \in V$ and forming an edge for any pair of vertices (i, j) for which $x_i + x_j > \theta$ holds for some threshold θ.

We generated threshold graphs by assigning an uniformly distributed binary value at random $b_i \in \{0, 1\}$ to each vertex $v_i \in V$ and construct G by repeatedly adding an isolated vertex v_i and connecting it with all previously added vertices if $b_i = 1$. The vertex set can be split into a core ($b_i = 1$) and a periphery set ($b_i = 0$). We set $b_{|V|} = 1$ to ensure that the resulting graph is connected and define the core size to be $\frac{4}{10}|V| = \sum_{i \in V} b_i$, see Fig. 12(c) for an example with 500 vertices. Since we do not want to have a perfect threshold graph, we only keep each edge with a probability of 80%.

4.2 Edge Embeddedness Methods

We compare different methods which assign a weight $w : E \to \mathbb{R}_{\geq 0}$ to each edge $e = (u, v) \in E$ depicting its embeddedness. All these methods are then extended using our UMST approach to guarantee the connectivity, such that a layout can be computed from the resulting graph. We use the following approaches to assign a weight to the edges.

Random: Assigns uniform random weights, as base line.

Jaccard: Jaccard coefficient, $\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$, as proposed by Satuliri et al. [34].

Simmelian: Triadic Simmelian backbone, as proposed by Nick et al. [29].

Quadrilateral: Our quadrilateral Simmelian backbone, which accumulates triadic effects at an edge with quadrangles (Sect. 3).

Density: Metric by Auber et al. [1] accumulating densities of different subgroups in the local neighborhood.

Ground Truth: Knowledge of class membership in the synthetic network is used to assign directly a low value to inter-cluster edges and a high value to intra-cluster edges.

4.3 Quality Metrics

In contrast to the synthetic networks there is no ground truth available for the Facebook networks. This makes it hard to evaluate outcomes of the different methods. Nevertheless, it was found that for many of the Facebook networks, the housing structure (dormitory attribute) is highly relevant for the underlying formation of social relations [29, 40]. We, therefore, use the dormitory attribute as a reference for evaluation.

Assume that we know the ground truth, meaning the class membership $c(v)$ of each vertex. A *perfect* algorithm, for example, would first remove all inter-cluster edges before starting to remove intra-cluster edges while obeying the required sparsification ratio. Since inter-cluster edges are removed priorly,

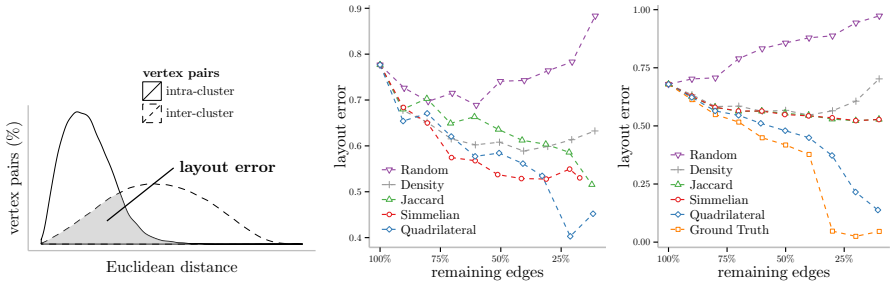
this increases the ratio between intra-cluster or homophily edges and the total number of edges.

If the edge embeddedness methods perform similar to this, the ratio of homophily edges

$$\text{homophily}(G) = \frac{\#\text{homophily edges}}{\#\text{homophily edges} + \#\text{heterophily edges}}$$

should monotonically increase, while gradually removing edges from the network according to their weight. Edges for which the class membership (attribute) of at least one vertex is missing are neglected.

Additionally, we would like to see how well this class membership is reflected in the layouts. Vertex pairs of the same class should have a small Euclidean distance, while pairs of different classes should have a large Euclidean distance. Looking at the curve of the Euclidean distance distribution of the intra-cluster and inter-cluster vertex pairs in Fig. 7(a), we define the layout error as the intersection area of these two curves. The layout error can also be interpreted as the percentage of vertex pairs, where the distinction whether they are in the same cluster or not cannot clearly be made based on the Euclidean distance. Since the computation of this quality metric is very time intensive, it was not feasible to analyze all Facebook100 networks with it.



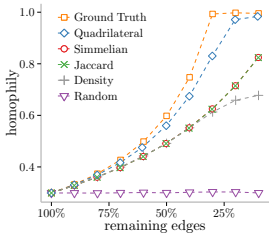
(a) layout error (b) Caltech36 (c) synthetic networks

Figure 7: Layout error of different edge embedding methods combined with UMST for (b) a real world network and (c) synthetic networks. (a) shows the layout error for a single point of the line chart in (b).

4.4 Results and Discussion

An interesting observation from Fig. 8 is that Jaccard and Simmelian perform very similar for most Facebook networks. Our method (Quadrilateral) clearly manages to distinguish between the different types of edges better than the other methods, especially in earlier phases of the sparsification.

For all 100 Facebook networks, the difference in homophily between Simmelian and Quadrilateral is shown by the length of a vertical segment in Fig. 9.



(a) Synthetic network model (PPM) with hidden homophily structure. Quadrilateral comes very close to the ground truth in distinguishing between intra- and inter-cluster edges.

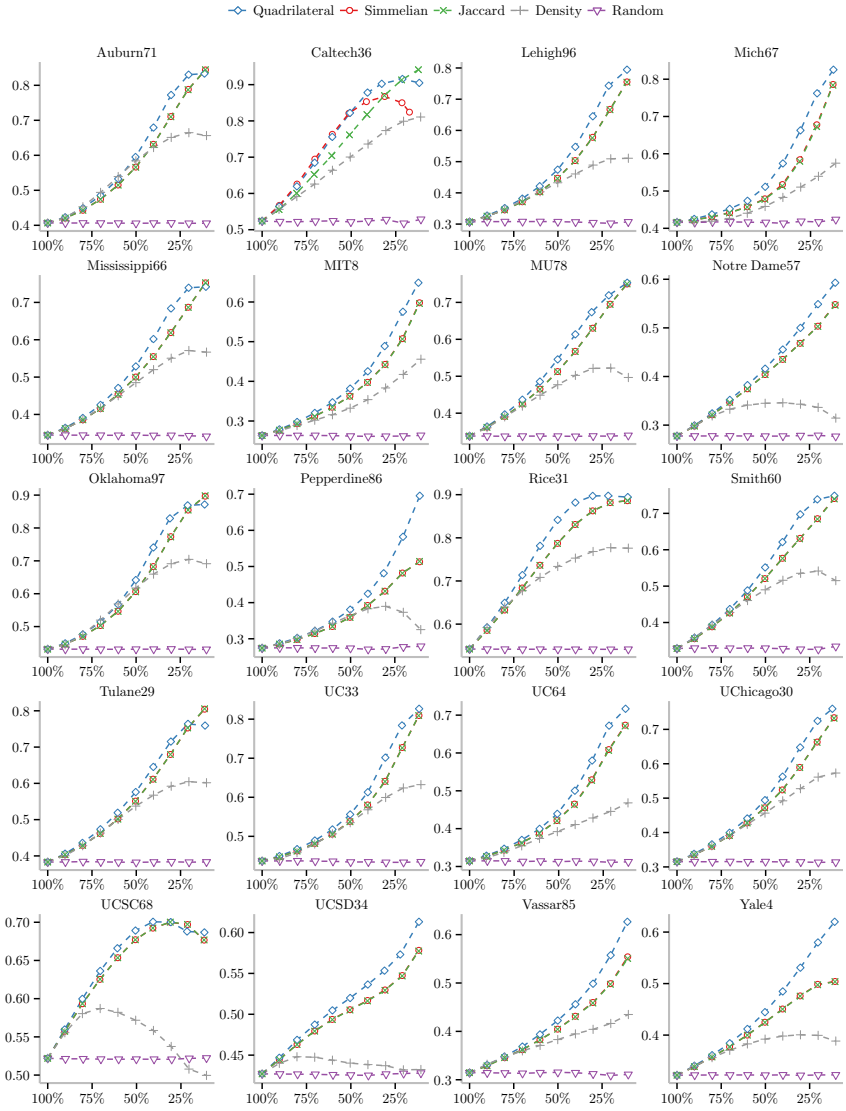


Figure 8: (b) Top 20 Facebook networks with high homophily structure in original network. Homophily (y-axis) is plotted against the number of remaining edges (x-axis). Overall Quadrilateral performs better than the others.

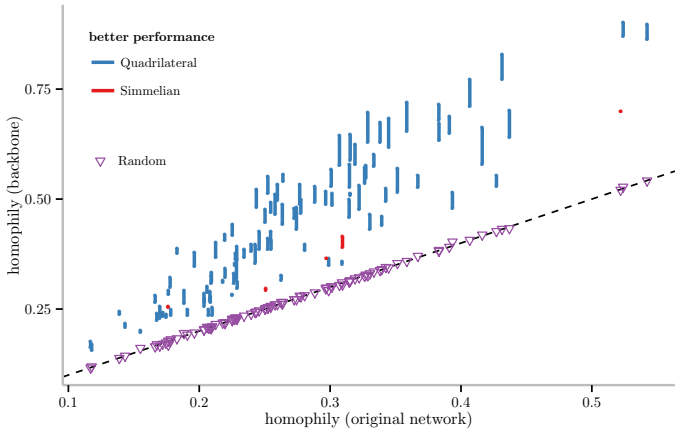


Figure 9: Dormitory-homophily of different backbones, with sparsification ratio 70%, (y-axis) compared to the homophily in the original network (x-axis) for all Facebook100 networks. Points above/below the dashed line indicate homophily increase/decrease respective the original network. Simmelian and Quadrilateral homophily values for corresponding networks have been connected by colored segments comparing their performance.

While both approaches increase the percentage of homophily edges (all segments above the diagonal dashed line), Quadrilateral clearly performs better, especially for networks with a higher percentage of homophily edges.

Although the homophily of Jaccard and Quadrilateral is nearly the same for the last but one step of the Caltech36 network (Fig. 8(b)), the Quadrilateral embedding creates the superior layout, as can be seen by the lower layout error in Fig. 7(b) or the drawings in Fig. 1(e) and 1(d). Furthermore, for the synthetic networks (PPM), Quadrilateral comes very close to the ground truth (Fig. 7(c)).

Figure 10 shows the layout error for four Facebook networks and the three best performing edge metrics (according to homophily). The layout clearly improves for the Rice and Smith network, but not much for the other two. One possible explanation for this could be that the dormitory attribute is not the explanatory variable for the formation of social relations in these two networks. Other attributes, as the expected year of graduation, can also explain parts of the revealed group structue, as can be seen in Fig. 11 for the Pepperdine86 network.

One can also observe in the final drawings that Jaccard keeps the clusters connected to a single center in multiple radial layers, while Quadrilateral expands the clusters more clearly, see Fig. 14 and Fig. 15.

The effectiveness of our layout quality metric is substantiated by the drawings in Fig. 1(c) and 1(d). In the latter many clusters, as light green and light blue, are more clearly visible. For the synthetic networks Quadrilateral comes very close to the ground truth, in terms of layout error (Fig. 7(c)). This finding is also supported by the drawings (Fig. 13) of a synthetic network.

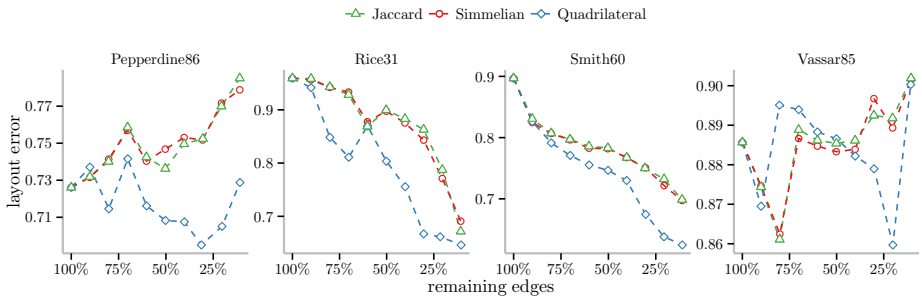


Figure 10: Layout error of Facebook networks w.r.t. the dormitory attribute. While improvement is not clear for Pepperdine86 and Vassar85, the layout is improved a lot for the networks with high homophily (Rice31 and Smith60).

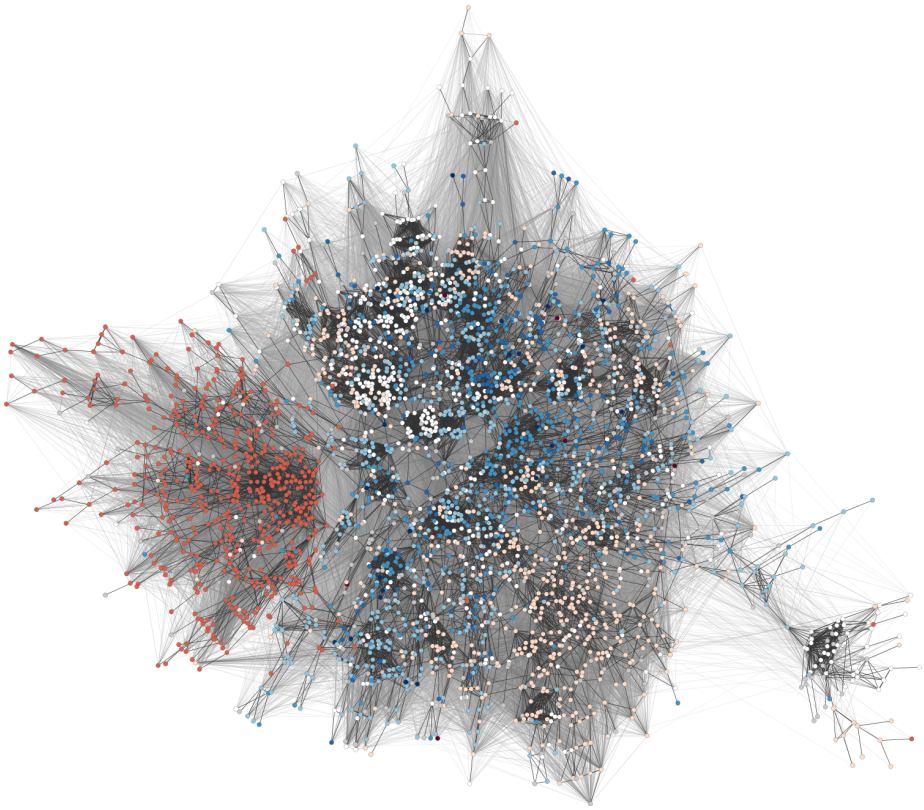
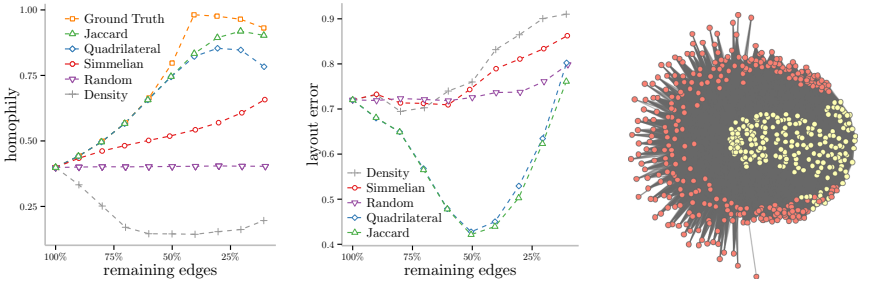
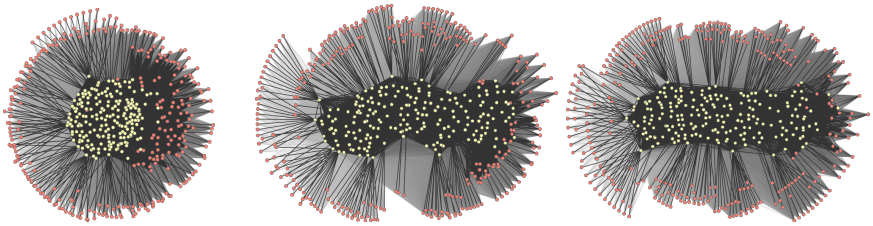


Figure 11: Drawing of the Pepperdine86 network. The *year* attribute is mapped to vertex color using interpolation (blue-white-red). Vertices with missing values are colored gray.



(a) structural quality (b) layout quality (c) extreme core-periphery network



(d) Quadrilateral, 50% (e) Quadrilateral, 20% (f) Jaccard, 20%

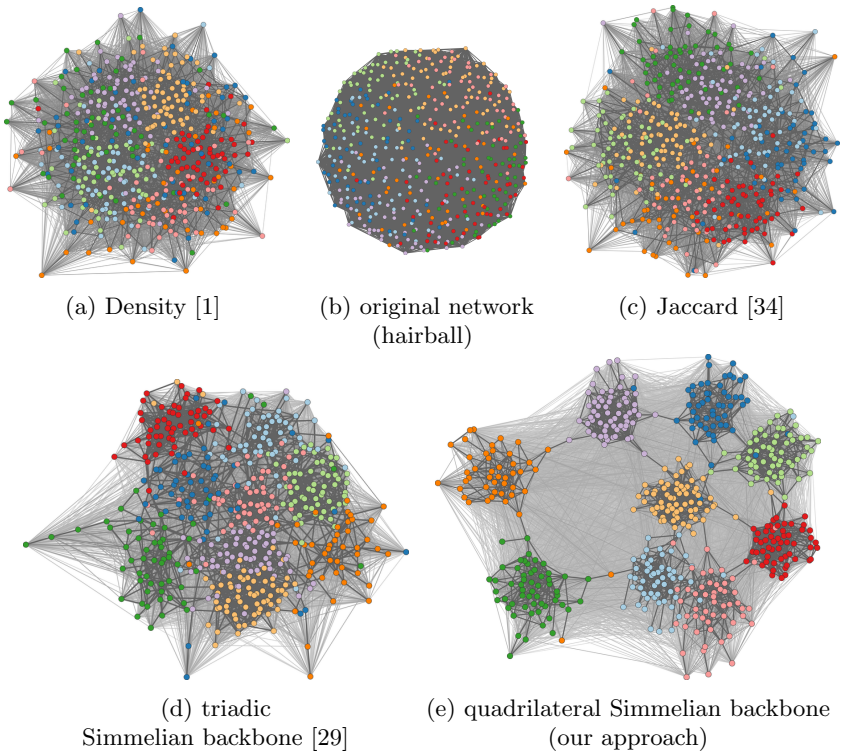
Figure 12: Threshold graph with extreme core-periphery structure (yellow-red). Untangling the hairball stretches the core due to the skewed connectivity of the periphery. Layout error increases due to skewed elliptic core shape.

For the threshold graph, Jaccard performs slightly better than Quadrilateral according to homophily, cf. Fig. 12(a), yet the layout error is nearly the same, see Fig. 12(b). The increase of the layout error for less than 50% remaining edges can be explained by the skewed elliptic core shape (Fig. 12(e) and 12(f)), which is a characteristic of the threshold graph structure. The backbone layout of the world trade network can be seen in Fig. 16. The core, mostly consisting of the countries with a large GDP, is separated from the periphery, based on the network structure only.

However, besides separating the core from the periphery our backbone approach is of limited use for these types of networks, as the low structural variation within the core does not allow further disassembly.

5 Conclusion

We proposed a sparsification approach to draw hairball graphs as encountered in online social networks. It is based on the idea that pairwise distances (the "degrees of separation") need to be increased without disrupting tightly-knit groups. The deeply embedded edges that such groups are made of are identified using a modified Simmelian backbone [29], and overall layout organization is stabilized by maintaining connectedness via the union of all maximum spanning trees.



(a) Density [1] (b) original network (hairball) (c) Jaccard [34]

(d) triadic Simmelian backbone [29] (e) quadrilateral Simmelian backbone (our approach)

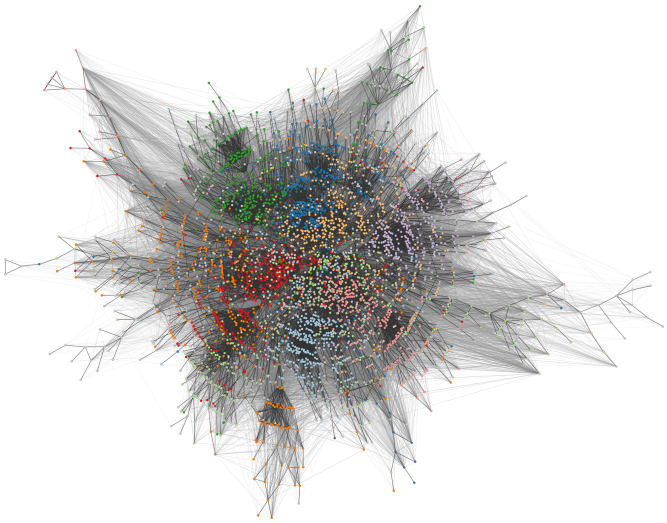
Figure 13: Backbone layouts of the same synthetic network determined by different edge embeddedness methods combined with UMST (20% remaining edges). Colors encode groups – ground truth, but have not been utilized by any of the methods.

An evaluation with empirical and generated networks showed that our novel metric manages to reveal relations deeply embedded in latent primary groups. In the resulting drawings such groups are separated from each other but still positioned in their global context. On the Facebook100 dataset, average distances increased from about 3 in the original friendship networks to about 14 in the backbone, thus easing the layout task for force-directed algorithms.

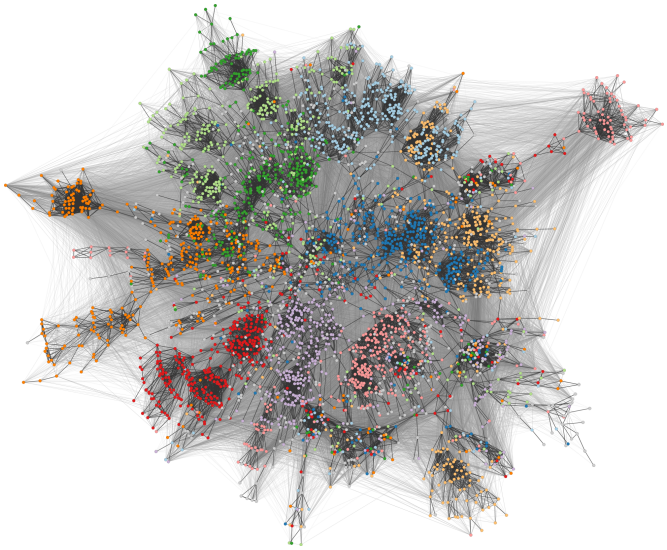
Our novel quadrilateral edge embeddedness metric proved to be more effective than previous approaches with respect to improving layout quality by way of amplifying homophily. It is thus likely to be useful as a preprocessing step for graph clustering algorithms as well.

Although our approach separates the core from the periphery in core-periphery networks, the drawings obtained for single-centered networks are rather inappropriate. By design, our technique appears to be best suited for small-world networks with multiple centers. While these are common, especially in social media, it will be interesting to identify variants for core-periphery and hierarchically clustered graphs.

In our illustrations we focused on emphasizing overall variation in density and how it is determined by backbone edges, yet drawing non-backbone edges still causes clutter. While this clearly shows the complexity of the original graph, alternative representations for these edges need to be explored and interactive filtering techniques might be beneficial for specific tasks.



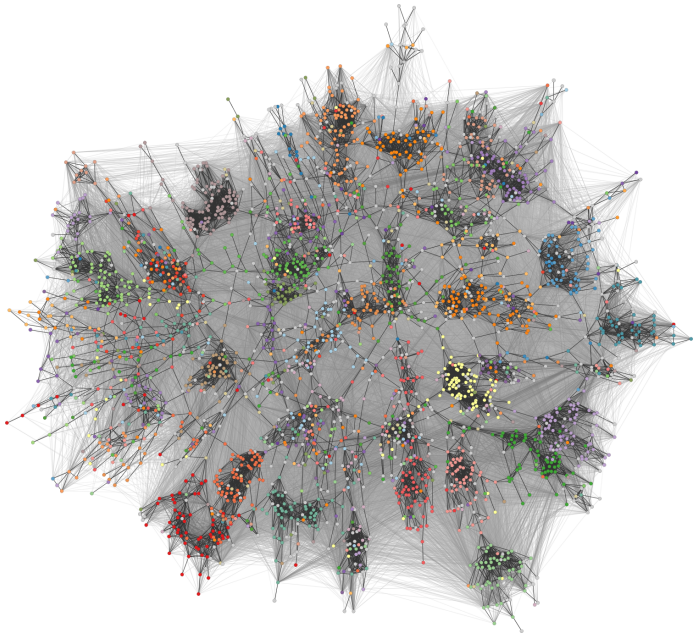
(a) Jaccard [34] with UMST



(b) our quadrilateral Simmelian backbone with UMST

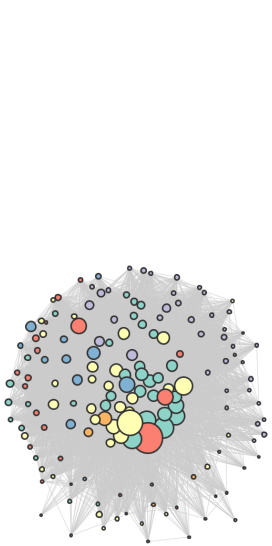
Figure 14: Drawings of Rice31 from the Facebook100 dataset with 4083 vertices and 10% of the 184K edges, using different edge embeddedness methods. Color encodes dormitory attribute, but has not been utilized by the drawing algorithm.

(a) Jaccard [34] with UMST

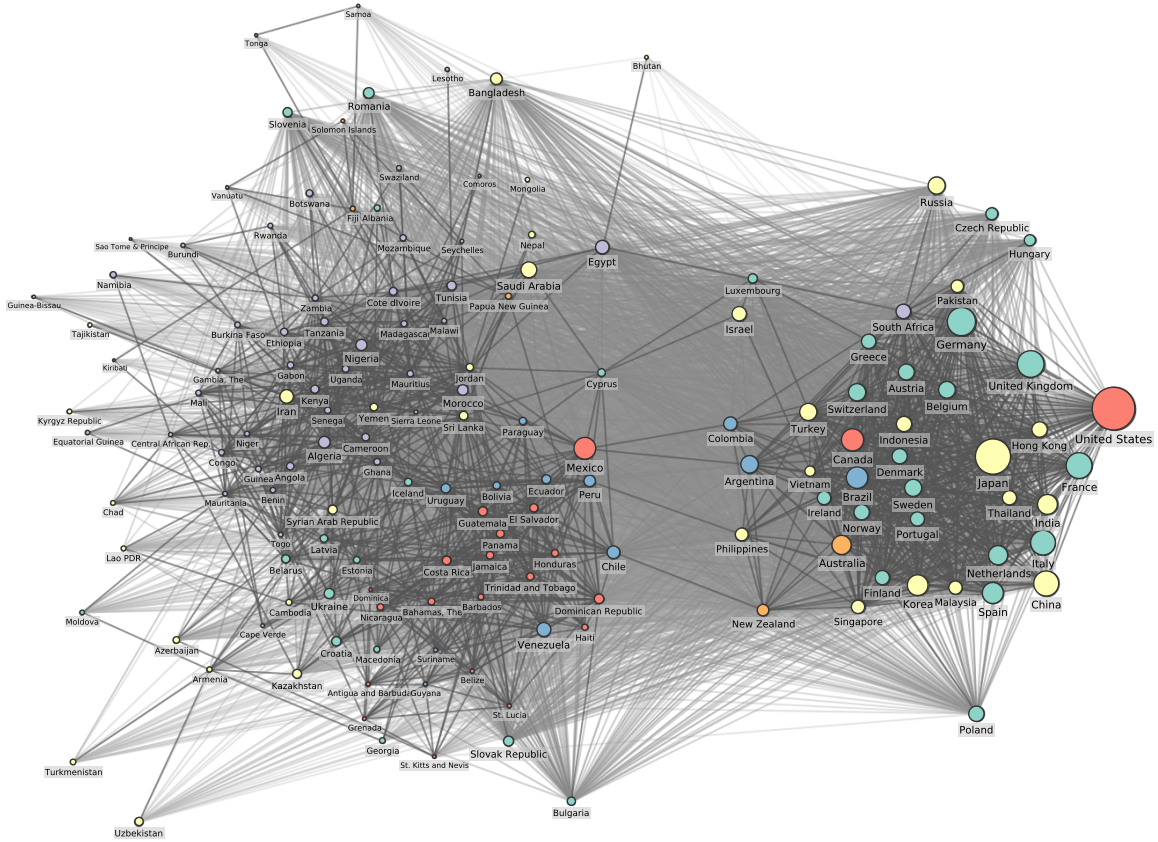


(b) our quadrilateral Simmelian backbone with UMST

Figure 15: Drawings of Smith60 from the Facebook100 dataset with 2970 vertices and 10% of the 97K edges, using different edge embeddedness methods. Color encodes dormitory attribute, but has not been utilized by the drawing algorithm.



(a) drawing, original network



(b) drawing, based on quadrilateral Simmelian backbone (20%)

Figure 16: (a) World trade network [39] with a core-periphery structure. (b) Backbone layout, which separates the core (right) from the periphery (left) based on graph structure only. The node size and color encode the GDP and the continent, respectively.

References

[1] D. Auber, Y. Chiricota, F. Jourdan, and G. Melançon. Multiscale visualization of small world networks. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS 2003)*, pages 75–81. IEEE Computer Society, 2003.

[2] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 47–55. ACM, 1996. doi:10.1145/237814.237827.

[3] K. Boitmanis, U. Brandes, and C. Pich. Visualizing internet evolution on the autonomous systems level. In S. Hong, T. Nishizeki, and W. Quan, editors, *Proceedings of the 15th International Symposium on Graph Drawing (GD 2007)*, volume 4875 of *Lecture Notes in Computer Science*, pages 365–376. Springer, 2007. doi:10.1007/978-3-540-77537-9_36.

[4] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *Proceedings of the 14th International Symposium on Graph Drawing (GD'06)*, volume 4372 of *Lecture Notes in Computer Science*, pages 42–53. Springer-Verlag, 2007. doi:10.1007/978-3-540-70904-6_6.

[5] U. Brandes and C. Pich. An experimental study on distance-based graph drawing. In *Proceedings of the 16th International Symposium on Graph Drawing (GD'08)*, volume 5417 of *Lecture Notes in Computer Science*, pages 218–229. Springer-Verlag, 2009. doi:10.1007/978-3-642-00219-9_21.

[6] R. S. Burt. Structural holes versus network closure as social capital. *Social capital: Theory and research*, pages 31–56, 2001.

[7] N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on Computing*, 14(1):210–223, 1985. doi:10.1137/0214017.

[8] J. Coleman. *Foundations of Social Structure*. Belknap Press, 1990.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[10] D. Dekker. Measures of Simmelian tie strength, Simmelian brokerage, and the Simmelianly brokered. *Journal of Social Structure*, 7(1), 2006.

[11] T. Dwyer, N. H. Riche, K. Marriott, and C. Mears. Edge compression techniques for visualization of dense directed graphs. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2596–2605, 2013. doi:10.1109/TVCG.2013.151.

[12] D. Eppstein and E. S. Spiro. The h-index of a graph and its application to dynamic subgraph statistics. *Journal of Graph Algorithms and Applications*, 16(2):543–567, 2012. doi:10.7155/jgaa.00273.

[13] E. R. Gansner, Y. Koren, and S. C. North. Graph drawing by stress majorization. In *Proceedings of the 12th International Symposium on Graph Drawing (GD '04)*, volume 3383 of *Lecture Notes in Computer Science*, pages 239–250. Springer-Verlag, 2005. doi:10.1007/978-3-540-31843-9_25.

[14] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[15] M. Granovetter. Economic action and social structure: The problem of embeddedness. *American Journal of Sociology*, 91(3):481–510, 1985.

[16] W. G. S. Hines. Geometric mean. In *Encyclopedia of Statistical Sciences*. John Wiley & Sons, Inc., 2004. doi:10.1002/0471667196.ess0877.pub2.

[17] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009. doi:10.1111/j.1467-8659.2009.01450.x.

[18] Y. F. Hu. Efficient and high quality force-directed graph drawing. *The Mathematica Journal*, 10:37–71, 2005.

[19] Y. F. Hu and L. Shi. Visualizing large graphs. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(2):115–136, 2015. doi:10.1002/wics.1343.

[20] C. Hurter, A. Telea, and O. Ersoy. Moleview: An attribute and structure-based semantic lens for large element-based plots. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2600–2609, 2011. doi:10.1109/TVCG.2011.223.

[21] T. J. Jankun-Kelly, T. Dwyer, D. Holten, C. Hurter, M. Nöllenburg, C. Weaver, and K. Xu. Scalability considerations for multivariate graph visualization. In *Multivariate Network Visualization*, volume 8380 of *Lecture Notes in Computer Science*, pages 207–235. Springer, 2013. doi:10.1007/978-3-319-06793-3_10.

[22] S. G. Kobourov. Force-directed drawing algorithms. In *Handbook of Graph Drawing and Visualization*, pages 383–408. Chapman & Hall/CRC, 2013.

[23] N. V. Mahadev and U. N. Peled. *Threshold graphs and related topics*, volume 56. Elsevier, 1995.

[24] R. N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems*, 11(1):193–197, 1999. doi:10.1007/s100510050929.

[25] F. McSherry. Spectral partitioning of random graphs. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537. IEEE Computer Society, 2001. doi:10.1109/SFCS.2001.959929.

[26] G. Melançon and A. Sallaberry. Edge metrics for visual graph analytics: A comparative study. In *Proceedings of the 12th International Conference on Information Visualisation (IV '08)*, pages 610–615. IEEE Computer Society, 2008. doi:10.1109/IV.2008.10.

[27] F. J. Newberry. Edge concentration: A method for clustering directed graphs. *ACM SIGSOFT Software Engineering Notes*, 14(7):76–85, 1989.

[28] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004. doi:10.1103/PhysRevE.69.026113.

[29] B. Nick, C. Lee, P. Cunningham, and U. Brandes. Simmelian backbones: Amplifying hidden homophily in facebook networks. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13)*, pages 525–532. ACM, 2013. doi:10.1145/2492517.2492569.

[30] A. Nocaj and U. Brandes. Stub bundling and confluent spirals for geographic networks. In *Proceedings of the 21st International Symposium on Graph Drawing (GD 2013)*, volume 8242 of *Lecture Notes in Computer Science*, pages 388–399. Springer, 2013. doi:10.1007/978-3-319-03841-4_34.

[31] M. Ortmann and U. Brandes. Triangle listing algorithms: Back from the diversion. In *Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX 2014)*, pages 1–8. SIAM, 2014. doi:10.1137/1.9781611973198.1.

[32] J. L. Pfaltz. The irreducible spine(s) of undirected networks. In *Proceedings of the 14th International Conference on Web Information Systems Engineering (WISE 2013), Part (2)*, volume 8181 of *LNCS*, pages 104–117. Springer, 2013. doi:10.1007/978-3-642-41154-0_8.

[33] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA*, 101:2658–2663, 2004. doi:10.1073/pnas.0400054101.

[34] V. Satuluri, S. Parthasarathy, and Y. Ruan. Local graph sparsification for scalable clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 721–732. ACM, 2011. doi:10.1145/1989323.1989399.

[35] S. Schnettler. A structured overview of 50 years of small-world research. *Social Networks*, 31(3):165–178, 2009. doi:10.1016/j.socnet.2008.12.004.

[36] G. Simmel. *The sociology of Georg Simmel*, volume 92892. Simon and Schuster, 1950.

[37] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. doi:10.1137/080734029.

[38] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, pages 81–90. ACM, 2004. doi:10.1145/1007352.1007372.

[39] A. Subramanian and S.-J. Wei. The WTO promotes trade, strongly but unevenly. *Journal of International Economics*, 72(1):151 – 175, 2007. doi:10.1016/j.jinteco.2006.07.007.

[40] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review*, 53(3):526–543, 2011. doi:10.1137/080734315.

[41] M. Tumminello, T. Aste, T. Di Matteo, and R. Mantegna. A tool for filtering information in complex systems. *Proc. Natl. Acad. Sci. USA*, 102(30):10421–10426, 2005. doi:10.1073/pnas.0500298102.

[42] F. van Ham and M. Wattenberg. Centrality based visualization of small world graphs. *Computer Graphics Forum*, 27(3):975–982, 2008. doi:10.1111/j.1467-8659.2008.01232.x.

[43] F. Zaidi, A. Sallaberry, and G. Melançon. Revealing hidden community structures and identifying bridges in complex networks: An application to analyzing contents of web pages for browsing. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT '09)*, pages 198–205. IEEE, 2009. doi:10.1109/WI-IAT.2009.36.

[44] F. Zhou, S. Mahler, and H. Toivonen. Network simplification with minimal loss of connectivity. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010)*, pages 659–668. IEEE Computer Society, 2010. doi:10.1109/ICDM.2010.133.