Group-In-a-Box Layout for Multi-faceted Analysis of Communities

Eduarda Mendes Rodrigues[*], Natasa Milic-Frayling [†], Marc Smith[‡], Ben Shneiderman[§], Derek Hansen[¶]

[*] Dept. of Informatics Engineering, Faculty of Engineering, University of Porto, Portugal
eduardamr@acm.org
[†] Microsoft Research, Cambridge, UK
natasamf@microsoft.com
[‡] Connected Action Consulting Group, Belmont, California, USA
marc@connectedaction.net
[§] Dept. of Computer Science & Human-Computer Interaction Lab
University of Maryland, College Park, Maryland, USA
ben@cs.umd.edu
[¶] College of Information Studies & Center for the Advanced Study of Communities and Information
University of Maryland, College Park, Maryland, USA
dlhansen@umd.edu

***Abstract*—Communities in social networks emerge from interactions among individuals and can be analyzed through a combination of clustering and graph layout algorithms. These approaches result in 2D or 3D visualizations of clustered graphs, with groups of vertices representing individuals that form a community. However, in many instances the vertices have attributes that divide individuals into distinct *categories* such as gender, profession, geographic location, and similar. It is often important to investigate what categories of individuals comprise each community and vice-versa, how the community structures associate the individuals from the same category. Currently, there are no effective methods for analyzing both the community structure and the category-based partitions of social graphs. We propose *Group-In-a-Box* (GIB), a meta-layout for clustered graphs that enables multi-faceted analysis of networks. It uses the treemap space filling technique to display each graph cluster or category group within its own box, sized according to the number of vertices therein. GIB optimizes visualization of the network sub-graphs, providing a semantic substrate for category-based and cluster-based partitions of social graphs. We illustrate the application of GIB to multi-faceted analysis of real social networks and discuss desirable properties of GIB using synthetic datasets.**

Keywords—network visualization; group-in-a-box; layout; meta-layout; force-directed; communities; clustering; semantic substrates.

I. INTRODUCTION

Network structures appear in many contexts, from biological systems to communications networks. With the recent proliferation of social media services such as Twitter and Facebook, public awareness and usage of social network data have increased. This led to a growing need for comprehensible visualizations of complex networks that enable exploratory analysis.

One particularly important aspect of social network analysis is the detection of *communities*, i.e., sub-groups of individuals or entities that exhibit tight interconnectivity among the wider population. For example, Twitter users who regularly retweet each other's messages may form cohesive groups within the Twitter social network. In a network visualization they would appear as clusters or sub-graphs, often colored distinctly or represented by a different vertex shape in order to convey their group identity.

In addition to the clusters that emerge from the network structure, individuals in a social network are often divided into categories that reflect specific attributes. For example, members of the Twitter community may be categorized based on the number of followers they have, the location from which they tweet, or the date they joined Twitter. Such attributes may be useful to gain further insights about the community. Thus, in addition to detecting communities based on the network structure, it is important to enable analysis of the social graph along various attributes of the individuals. At the same time, none of the widely adopted layouts for visualizing clustered graphs accommodate such multi-faceted analysis.

Our research fills that gap by extending the work on network visualization with semantic substrates [20]. We introduce *Group-In-a-Box* (GIB), a meta-layout for clustered graphs that enables multi-faceted analysis of networks. GIB uses the treemap space filling technique to create graphs for individual categories, thus providing a semantic substrate for laying out each group. Furthermore, we demonstrate the use of the method to analyze communities, i.e., graph clusters by assigning them specific spatial regions, applying layout algorithms to reveal the local cluster structure, and observing the categories that their members fall into. In both cases, we arrive at well separated sub-graphs that clearly reveal the

intra-group network structures and the attributes of their members.

Our experiments illustrate the application of the GIB multi-faceted analysis to a social network that arises from the voting activities of U.S. Senators. Additionally, we use synthetic networks, generated according to pseudo-random, *small world* and *preferential attachment* network models, to demonstrate that the GIB layout exhibits desirable properties for visualizing vertices that connected different clusters. In social networks such individuals are often referred to as *boundary spanners* and serve as connectors, forming communication pathways through which information can be exchanged across communities. Thus, GIB is equally useful for gaining insights about the role of boundary spanners within their own community. Based on our experiments, we recommend generalization of the GIB meta-layout to accommodate multi-faceted analysis involving hierarchical classification schema and hierarchical clustering algorithms.

In the following sections, we introduce the graph clustering and layout algorithms used in our study and describe the GIB layout in detail. Through empirical observations of specific datasets and network visualization methods, we explore various aspects of the multi-faceted analysis. We conclude with the discussion of related work and guidelines for devising algorithms for multi-faceted analyses, including the need for readability criteria to optimize the utility of the GIB-like layouts.

II. Background and Problem Definition

Community detection algorithms aim to identify cohesive, tightly connected sub-graphs within a network. These structural clusters can be analyzed by considering both intra-cluster and inter-cluster links. Next, we introduce the graph clustering and layout algorithms that we used in our experiments and analysis.

A. Detecting Communities through Network Clustering

The notion of a community refers to a subset of vertices among which the distribution of edges is denser than with other vertices in the network. There are numerous methods for detecting communities, including hierarchical clustering, graph partitioning by maximizing specific criteria such as network modularity, and similar.

The *Girvan–Newman* algorithm [7] takes a divisive hierarchical approach. It uses *edge betweenness* as weights in the divisive process. Community boundaries are detected by progressively removing from the network the edges with the highest betweenness, re-calculating the betweenness of the remaining edges at each step. If a network contains loosely connected communities, e.g., linked by a few inter-cluster edges, then all shortest paths must pass through one of these few edges. Thus, such inter-cluster edges will have high betweenness and the associated vertices will have high *vertex betweenness centrality* [8], a useful measure of the vertex' influence on the information flow, especially in networks where such flows primarily follow the shortest available path.

The *Clauset-Newman-Moore* (CNM) algorithm [3], on the other hand, is a hierarchical agglomeration algorithm for detecting community structure, which is computationally more efficient than Girvan-Newman and similar algorithms. Its running time for a network with n vertices and m edges is $O(md \log n)$, where d is the depth of the hierarchy. However, in practice, the application of CNM is limited to medium-size networks, up to half a million vertices. Wakita and Tsurumi [23] show that this inefficiency is caused by merging communities in an unbalanced manner. They propose simple heuristics to achieve a balanced merge and scale up to social networks with millions of users.

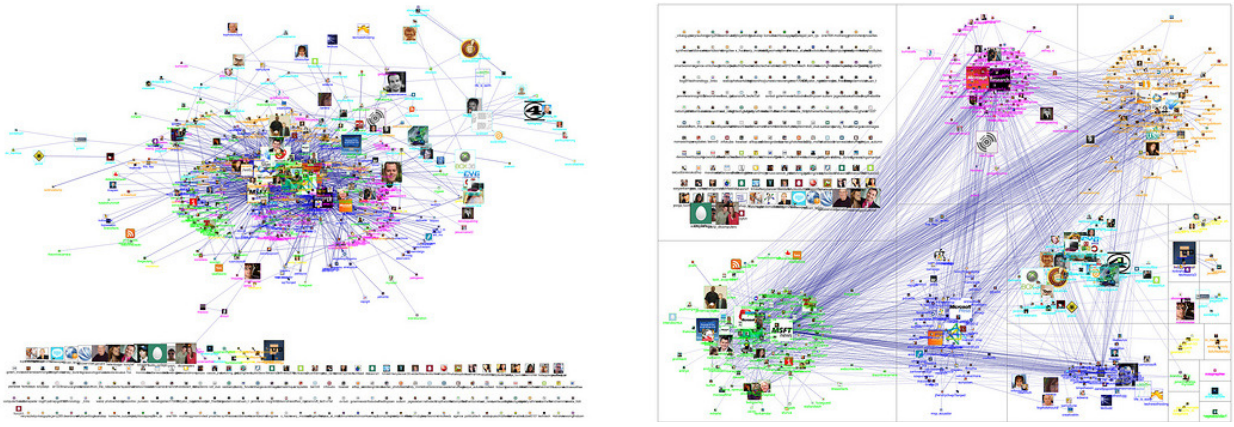
B. Graph Layouts

A common approach to visualizing networks is to use *force-directed* layout of vertices and edges based on models of physical systems in order to arrive at optimal graph structures. The *Fruchterman-Reingold* (FR) layout [9] belongs to that class of algorithms. It treats vertices as steel rings and edges as springs between them, observing two forces: an attractive force that is analogous to a spring force abiding by Hooke's law and a repulsive force that acts as an electrical force between charged particles, similar to Coulomb's law. The algorithm minimizes the energy of the system by moving the vertices and changing the forces between them until the system reaches an equilibrium state. This layout algorithm is useful for visualizing large undirected networks and, generally, creates overall satisfying layouts, placing vertices of the same cluster in the proximity of each other. However, some local areas of the graph may still be sub-optimally laid out.

Harel and Koren (HK) proposed a fast multi-level graph layout algorithm to achieve better visualizations [11]. Their approach involves a two-phase method that recursively coarsens the graph to arrive at its multi-level representation. First, the graph is embedded in a high dimensional space and then projected onto a 2-D plane using principal components analysis.

C. Faceted Analysis of Communities

In practice, social network analysis (SNA) is multi-faceted. Often it is important to analyze the roles of individuals that arise from their interaction within and across communities. However, in order to gain further insights, SNA may explore additional, non-structural attributes that are not considered by the clustering algorithm. Indeed, the individuals may belong to different categories based on gender, profession, age, location, etc. It is often useful to observe the distribution of these attributes across the community structure. However, to support the analysis that takes into account both the detected community structure and the categories that individual belong to, we need to address two specific issues:



(a) (b)

Figure 1. (a) Harel-Koren (HK) fast multi-scale layout of a clustered network of Twitter users, using color to differentiate among the vertices in different clusters. The layout produces a visualization with overlapping cluster positions. (b) Group-in-a-Box (GIB) layout of the same Twitter network: clusters are distributed in a treemap structure that partitions the drawing canvas based on the size of the clusters and the properties of the rendered layout. Inside each box, clusters are rendered with the HK layout.

1) There are no layouts that enable flexible visualizations of categories and links across categories that arise from the underlying network structure.

2) Clustering and layout algorithms are typically applied separately. As a result, the graph layout does not take into account cluster membership of vertices and causes two effects: (i) occlusion of clusters within the graph layout and (ii) loss of information about the structure of individual clusters and inter-cluster connectivity.

The former is illustrated in Figures 1 (a) and Figure 6 (a) where the CNM clustering algorithm is combined with the HK layout. The clusters overlap, as indicated through 'color bleeding' throughout the network visualization.

III. GIB Layout Algorithm

We propose to support multi-faceted network analysis by generalizing the approach of network visualization with semantic substrates [20]. We present the *Group-In-a-Box* (GIB) layout that simultaneously captures clustering and categorization of vertices in a network graph. The GIB algorithm,

- Uses the treemap space filling technique [1], [15], [19] to partition the graph canvas into regions of varying sizes, in which individual clusters or category groups are displayed.
- Allows a choice of layout algorithms for optimizing the layout of the sub-graphs within each region.
- Enables assignment of visual properties to vertices and edges within and across clusters and category groups.

We choose the treemap approach for the rich information it conveys about the relative size of individual clusters or categories (e.g., see Figure 1 (b)) and for its extensibility to hierarchical structures. Indeed, treemaps are designed to represent trees through nested rectangular regions. A rectangular area is subdivided into a set of rectangles that represent the top level vertices in the tree hierarchy. This process continues recursively, creating rectangles that represent each level in the tree through alternate vertical and horizontal subdivision of the rectangles.

GIB was implemented as part of the NodeXL network analysis tool[1]. NodeXL supports computation of network metrics, such as closeness and betweenness centrality, and mapping of vertex properties onto visual characteristics, such as size, color, and shape of the vertices [10][21]. We use NodeXL to generate all the graphs presented in the paper.

IV. GIB Method for Multi-Faceted Analysis

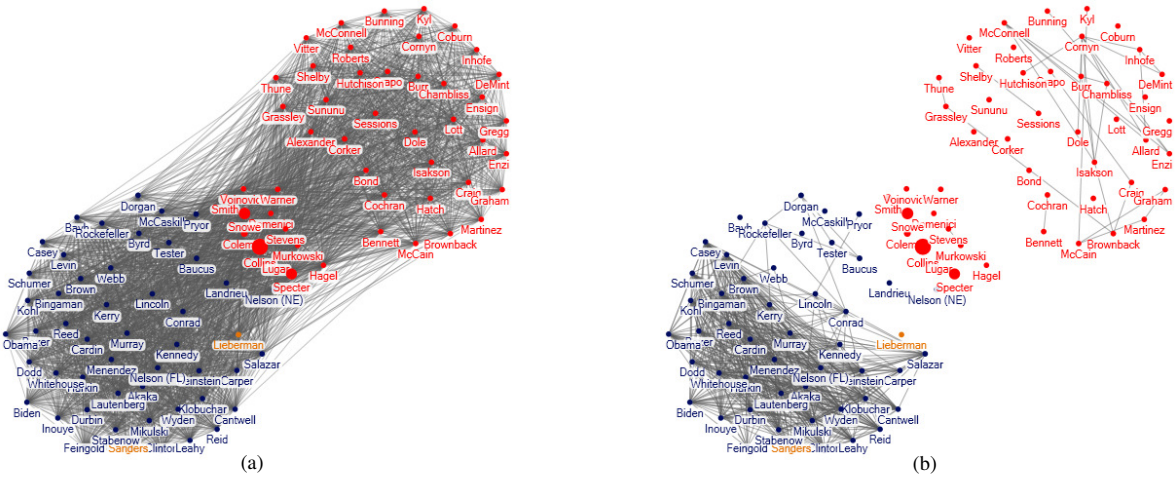
In this section we demonstrate the use of GIB in multi-faceted analysis of the U.S. Senate voting patterns in 2007[2].

A. Dataset

The *U.S. Senate co-voting network* was created from the data that connects senators to one another based on the number of times they voted the same way (i.e., both in favor or both against a bill). The network graph includes 98 vertices and 9506 weighted edges, each representing the percentage of voting agreement between a pair of senators. The graph is complete, i.e., all the senators are connected to each other. In order to capture strong relationships among the senators, we focus on edges between senators whose level of agreement is above 50%. The dataset also includes attributes about the senators, such as name, party affiliation, the state they represent, and the number of votes.

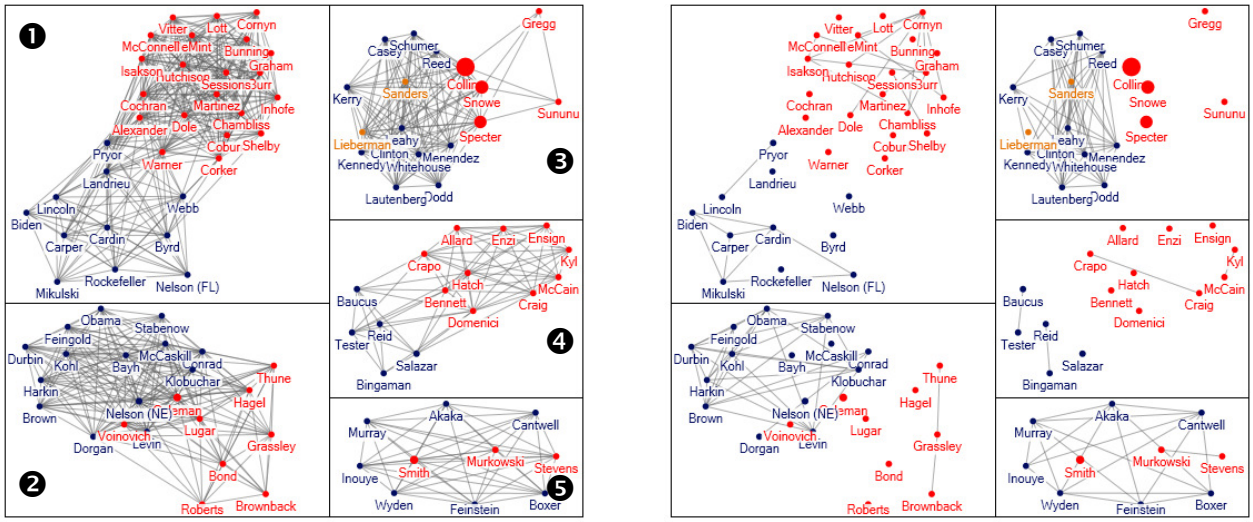
[1] NodeXL is freely available at http://nodexl.codeplex.com.

[2] Data originally provided by Chris Wilson of *Slate* magazine available in the NodeXL template format at http://casci.umd.edu/NodeXL_Teaching.



(a) (b)

Figure 2. The 2007 U.S. Senate co-voting network graph, obtained with the Fruchterman-Reingold (FR) layout. Vertices colors represent the senators' party affiliations (blue: Democrats; red: Republicans; orange: Independent) and their size is proportional to betweenness centrality. Edges represent percentage of agreement between senators: (a) above 50%; (b) above 90%.



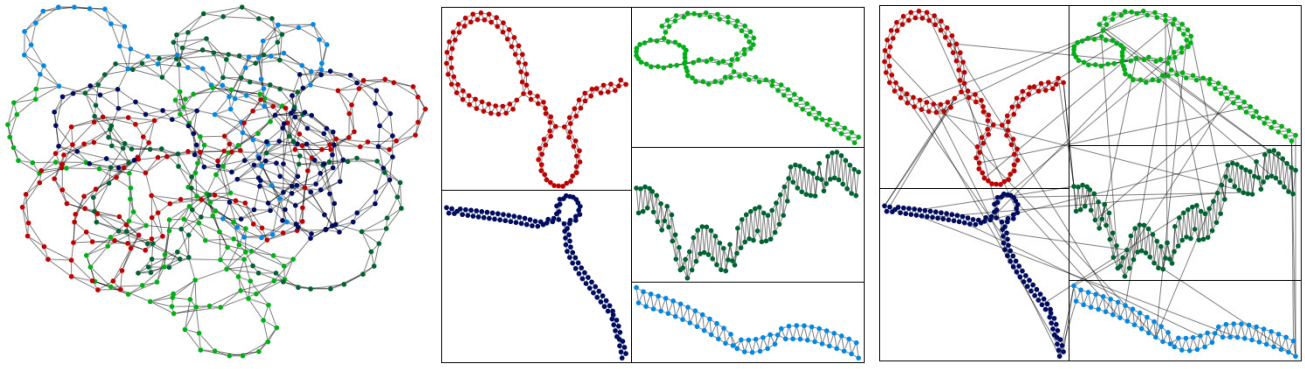
(a) (b)

Figure 3. The 2007 U.S. Senate co-voting network graph, visualized with the GIB layout. The group in each box represents senators from a given U.S. region (1: South; 2: Midwest; 3: Northeast; 4: Mountain; 5: Pacific) and individual groups are displayed using the FR layout. Vertices colors represent the senators' party affiliations (blue: Democrats; red: Republicans; orange: Independent) and their size is proportional to betweenness centrality. Edges represent percentage of agreement between senators: (a) above 50%; (b) above 90%..

B. GIB Application to U.S. Senate Co-voting Network

The FR layout of the U.S. Senate co-voting network yields the graph shown in Figure 2 (a). In this visualization, the vertices are colored according to the senators' party affiliation and sized according to the vertex betweenness centrality. Further filtering of the edges, to retain those with weight above 90%, yields the graph shown in Figure 2 (b), showing a strong voting agreement among the senators of the Democratic party.

Although this analysis already reveals interesting facts about the party cohesion and boundary spanners (represented by larger vertices), one may want to explore additional facets of the data to gain further insights about the voting patterns. We use GIB to perform such analysis. First, we group the States into five regions: Northeast, Midwest, South, Mountain, and Pacific. We then use these regions to divide senators into five groups, one group per region. Using the same edge filtering thresholds as before, the GIB layout results in Figure 3. GIB reveals further information about the cohesiveness of the two parties across different U.S. regions. For example, the Republican senators exhibit more voting cohesion in the southern states than the Democrats. Such observation was not possible in the full graph visualization.



(a) (b) (c)

Figure 4. Small-world network graph visualization obtained with the Harel-Koren layout, after clustering the graph with the Clauset-Newman-Moore community detection algorithm (5 clusters). (a) Full graph with 500 vertices colored according to the cluster membership. (b) GIB showing the structural properties of the individiual clusters. (c) GIB layout of the same 5 clusters showing inter-cluster edges.

V. GIB Application to Community Clusters

The GIB layout can also be applied to gain further insights into clustered graphs, in particular about the boundary spanners whose position within the original graphs can be obscured due to cluster occlusion. By generating synthetic network datasets, with known structural properties, we illustrate the conditions under which the GIB layout leads to new insights.

A. *Datasets*

We used the CONTEST Matlab toolbox [22] to generate the following synthetic network datasets:

- *Small-world network* with 500 vertices (avg. shortest-path length = 10.349; clustering coefficient = 0.467), generated according to the Watts-Strogatz model [24].
- Two *pseudo-random graphs*, each containing 5 clusters of different size (comprising 20, 40, 60, 80 and 100 vertices), generated according to the Erdös-Rényi model [6], with intra-cluster edge probability of 0.15. The two graphs differ in the inter-cluster edge probability set to 0.005 and 0.05, respectively.
- *Scale-free network* comprising 500 vertices, generated using the Barabási and Albert preferential attachment model [2]. Each new vertex was given 3 edges on arrival, leading to a scale-free degree distribution.

B. *GIB Visualization of a Small-world Network*

Small-world networks are characterized by high clustering coefficients and small average shortest path between pairs of vertices. Vertices in such network are densely connected with only a small number of neighbors, but can be reached from any other vertex in the network through a short path. The HK layout of the clustered *small world network* shown in Figure 4 (a) while enabling the observation of some structural regularity, it obfuscates the details about the structure of individual clusters detected by the CNM algorithm. On the other hand, the GIB visualizations shown in Figures 4 (b) and (c) isolate each of the five clusters into its own box, increasing the visibility of intra-cluster (b) and inter-cluster (c) edge structures. GIB provides clarity regarding the regular clique structures of the local neighborhoods within each cluster. It also enables the identification of vertices with "long range" edges, responsible for shrinking the network diameter.

C. *GIB Visualization of Pseudo-random Graphs*

The pseudo-random graphs containing synthetic clusters do not exhibit any particularly interesting intra-cluster structure to analyze. However, they enable us to explore limitations regarding the visualization of clustered graphs for a range of inter-cluster edge probabilities. Additionally, they enable us to observe the positioning of boundary spanners both with and without GIB.

The visualizations in Figure 5 show two pseudo-random networks with five clusters displayed using the HK fast multi-scale layout algorithm. GIB is used to isolate each of the five clusters into its own box. Figures 5 (a) and (b) illustrate the effect of increasing the inter-edge density on the occlusion of the clusters and the re-positioning of boundary spanners (i.e. the vertices with the largest size), which tend to gravitate towards the periphery of the clusters, when these are more separated. The GIB layouts, shown in Figures (c) and (d), display essentially the same intra-cluster structure for both graphs. This is expected since both have the same intra-cluster edge probability of 0.15. In this case, GIB enables a clear visualization of the centrality of boundary spanners within their own community.

D. *GIB Visualization of a Scale-free Network*

The scale-free network shown in Figure 6 (a) models structural properties of real social networks with power-law degree distributions. The CNM algorithm identified 10 clusters in this network. The color of the vertices reflects the cluster membership while the size corresponds to the vertex betweenness centrality. By removing the boundary edges and applying the HK layout algorithm we arrive at the graph (b), which renders separate clusters on top of each other.

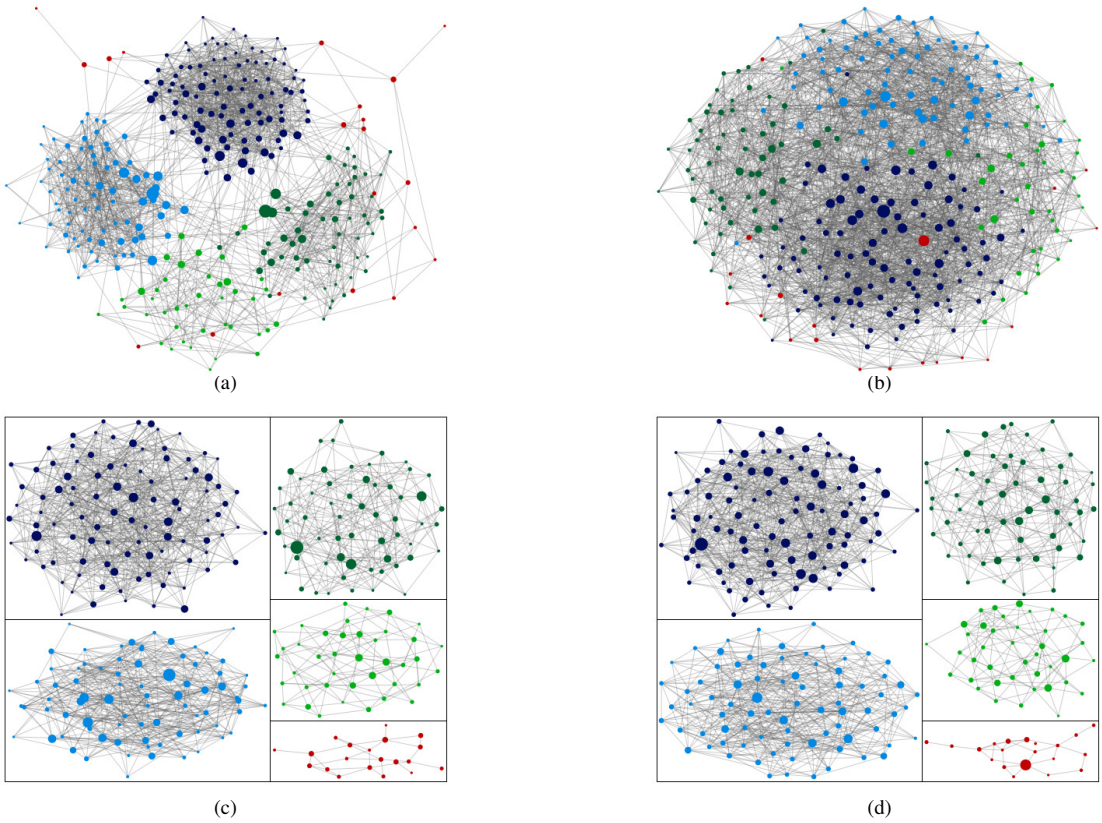


Figure 5. Pseudo-random graphs with 5 clusters of different sizes (comprising 20, 40, 60, 80 and 100 vertices), with intra-cluster edge probability of 0.15: (a) inter-cluster edge probability of 0.005; (b) inter-cluster edge probability of 0.05. The graphs are visualized using the Harel-Koren fast multi-scale layout agorithm and vertices are sized by betweenness centrality. The visualizations in (c) and (d) are the corresponding GIB layout.

The GIB algorithm provides an improved and clearer view of the structure of individual clusters. It also shows that vertices with high betweenness play a central role within individual clusters and are pulled towards the center of the clusters. This is expected in networks generated through preferential attachments since such vertices also exhibit high closeness centrality.

VI. Discussion and Related Work

A. *Observations from the GIB Experiments*

Investigation of GIB across real and synthetic networks revealed several properties. First, GIB layout subsumes three important aspects in the visualization of networks: (1) it enables simultaneous filtering or grouping of vertices based on a single or multiple attributes, (2) it enables layout optimization of individual groups while maintaining visual awareness of other groups, and (3) it supports explicit connection among close and remote vertices across groups. While some of these aspects can be achieved by simple filtering, the unifying effect that GIB offers is unique and not easily achieved by other methods.

Furthermore, with the focus on the local optimization of the group structure, GIB is robust and applicable across networks with different clusterability. Indeed, even in the case of graph visualizations without clearly discerned cluster boundaries, GIB adds value by showing the local, within-group link structure and inter-linking of the groups.

Our method of assessing the effectiveness of the GIB layout is based on observations how well GIB reveals prior knowledge about the networks. We also consider how well it abides by the design principle proposed by Krempel [17]: "*constrain the solutions to simple patterns, patterns which are known to put relatively little demand on the perceptual skills of an audience*." In that vein we considered relatively standard notions and characteristics of networks that the users are likely to observe and interpret by themselves.

Experiments with the synthetic networks show that GIB identifies and confirms known properties of specific graphs:

- The layout of small world networks highlights the long-range links between remote vertices. From inter-cluster links one can observe the cross-cluster effects of remote vertices (Figure 4).
- The layout of random networks shows boundary spanners pulled centrally towards the direction of other clusters. Locally, within their own cluster, they are uniformly distributed, as expected. That is the case, both when the links across clusters are sparse and when they are dense (Figure 5).
- In scale-free preferential attachment networks, the GIB local structure clearly highlights the presence of vertices with both high closeness and betweenness centrality. This

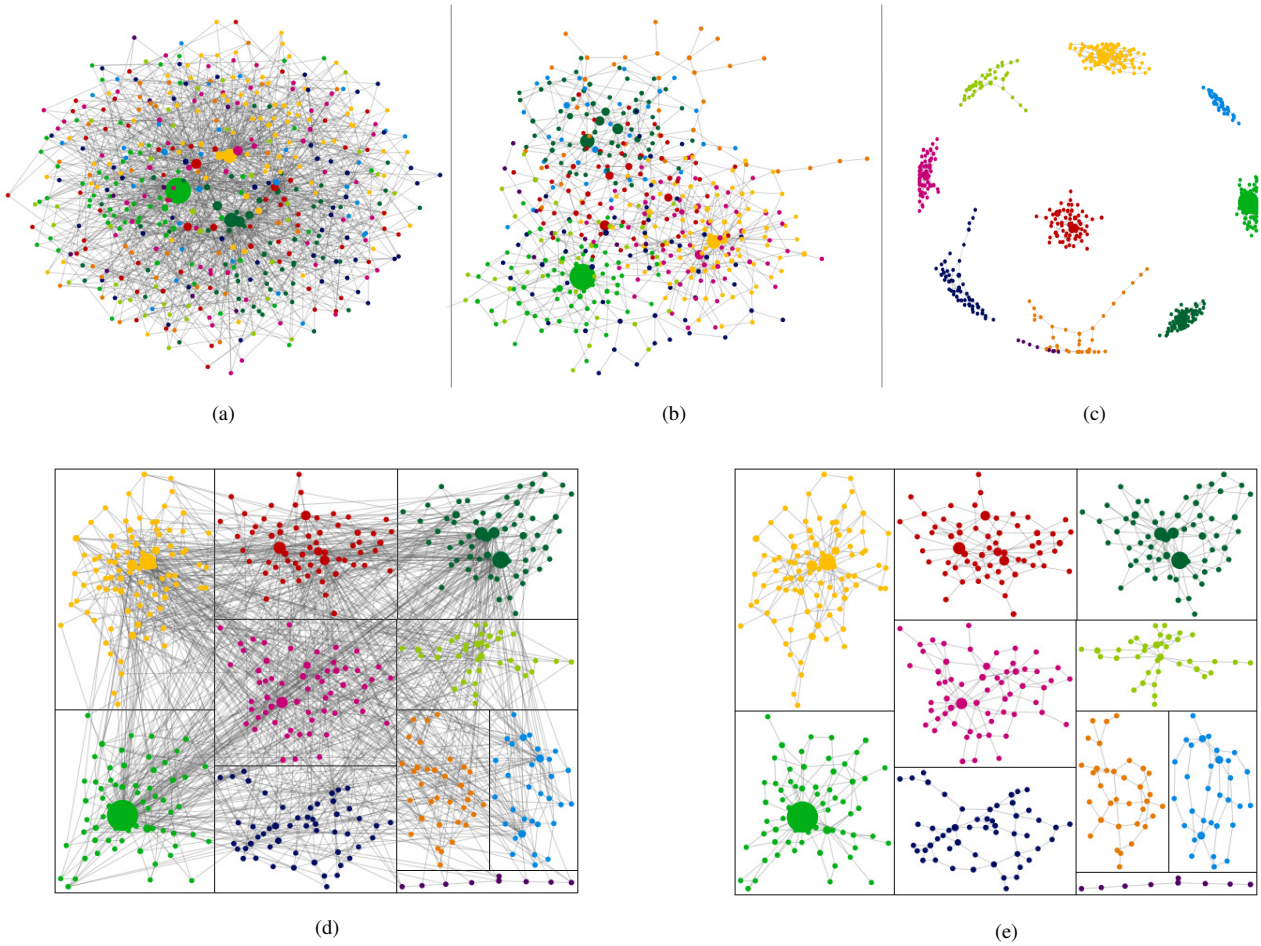


Figure 6. Scale-free network with 10 clusters detected by the Clauset-Newman-Moore algorithm. Vertices are colored by cluster membership and sized by betweeness centrality. (a) Harel–Koren layout of the clustered graph. (b) Harel–Koren layout after removing inter-cluster edges. (c) Fruchterman-Reingold layout after removing inter-cluster edges. (d) GIB showing inter-cluster edges and (e) GIB showing intra-clsuter edges.

is a defining property of such networks, giving advantage to vertices and groups that form earlier in the network lifecycle (Figure 6).

Thus, for a range of analysis scenarios, from multi-attribute characterization of networks to consistent insights about network properties, GIB appears as a promising tool for analyzing complex networks.

B. Related Work on Multi-facted Network Analysis

In their work on semantic substrates, Shneiderman and Aris [20] promote two principles: (1) the use of semantic substrates that are user-defined and (2) the use of controls, such as sliders, to mediate the visibility and clutter in visualization and ensure comprehensibility of the network graphs. We extend this approach to semantic groupings of nodes that can be expressed by a single or multiple attributes, as in the U.S. Senate voting analysis (Figures 2 and 3). We show that with GIB the user can explore semantic substrates that are based on either structural clusters or attribute-based groupings of graph vertices. These, in turn, can be further overlaid with additional attributes associated with vertices. In this sense, GIB expands the range of scenarios in which semantic substrates can be applied.

Harrer et al. [12] discuss various analyses of complex networks and illustrate how rich semantic underpinning, such as ontologies that support user interaction around the knowledge base, can lead to more complex network representations. In such cases, one is likely to see a shift in the network modeling, e.g., from a bi-partite graph analysis to multi-mode networks where more complex interactions need to be expressed. In this view, the GIB layout covers a limited set of scenarios, where attributes themselves do not induce additional structure that needs to be taken into account. Moving towards a hierarchical representation of categories would be the first step in that direction.

C. Related Work on Layout of Clustered Graphs

Clutter in network visualization may occur due positioning of the vertices or through interaction with the graph, e.g., by exposing labels on vertices or edges. This issue has been addressed by modifying the original layout to improve the readability. However, such transformations of

the graph layout need to be done cautiously to avoid distorting the structural aspects of a network. Eades and Feng [5], for example, address the issue of visualizing complex clustered structures by providing multiple levels of abstraction that preserve the *mental map* of the original graph. In another approach, Henry et al. [13] propose using vertex replication in order to emphasize the role of important actors in the community and reduce inter-cluster edge crossing [14]. However, while beneficial for a number of usage scenarios, this method does alter the network structure and may affect visual perception during analysis.

Other approaches take into account readability criteria by adding layout optimization constraints to the force-directed algorithms. For example, Kamps et al. [16] add a "vertex-vertex repulsion force" to the classical spring embedding model [4] in order to minimize the overlapping area of two vertices. Harel and Koren's method [11] calculates a repulsive force by considering the variant of the distance between the boundaries of vertex pairs. For clustered graphs, Noack [18] proposes optimization of planar layouts through minimization of the LinLog energy which takes into account the clustering properties of the network.

Within the multi-faceted network analysis, we apply a hybrid approach, promoting GIB in the contexts where the network structure and category groupings need to be jointly analyzed. The criteria for combining the two will evolve from the investigation of usage scenarios and developed user practices.

VII. Concluding Remarks and Future Work

In this paper we address the issue of supporting multi-faceted analysis of networks through an extension of the semantic substrate visualization approach. We introduce a novel graph layout algorithm, *Group-in-a-Box* (GIB), which enables simultaneous visualization of network clusters and categories of vertices in the network.

We demonstrate the benefits of the unified approach through experimentation with real and synthetic networks with diverse network properties. We show that the GIB layout is consistently effective in revealing the expected network structure in simulated data.

The current design of GIB will be further refined and expanded to support hierarchical clustering of networks and to incorporate support for taxonomy of categories. By observing how users apply GIB in different contexts, we expect to arrive at readability criteria for hybrid layouts like GIB. In multi-faceted analysis, the interconnection of two distinct groups, based on categories and clusters, needs to be supported in order to maximize the layout utility.

References

[1] Asahi, T., Turo, D. and Shneiderman, B. (1995) Using treemaps to visualize the analytic hierarchy process. *Information Systems Research*, 6(4): 357-375.

[2] Barabási, A.L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439): 509-512.

[3] Clauset, A., Newman, M.E.J. and Moore, C. (2004) Finding community structure in very large networks. *Physics Review E,* 70, 06111.

[4] Eades, P. (1984). A Heuristic for Graph Drawing. *Congressus Numerantium,* 42:149-160.

[5] Eades, P. and Feng, Q.-W. (1997). Multilevel Visualization of Clustered Graphs, *Proc. Graph Drawing, LNCS*, *1190*: 101-112.

[6] Erdös, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae (Debrecen)*, 6:290-297.

[7] Girvan, M. and Newman, M.E.J. (2002). Community structure in social and biological networks, *Proc. National Academy of Sciences*, 99:7821-7826.

[8] Freeman, L. C. (1977). A set of measures of centrality based on betweenness.*Sociometry*, 40:35-41.

[9] Fruchterman, T.M.J., and Reingold, E.M. (1991). Graph Drawing by Force-Directed Placement. *Software: Practice & Experience*, 21(11): 1129-1164.

[10] Hansen, D.L., Shneiderman, B., Smith, M.A. (2010). Analyzing Social Media Networks with NodeXL: Insights from a Connected World. Morgan Kaufmann.

[11] Harel, D. and Koren, Y. (2002). A Fast Multi-Scale Algorithm for Drawing Large Graphs. J. of Graph Algorithms Appl., 6(3):179-202.

[12] Harrer, A, Malzahn, N, Zeinin, S, and Hoppe, HU (2007). Combining Social Network Analysis with Semantic Relations to Support the Evolution of a Scientific Community, *Proc. 7th Computer Supported Collaborative Learning Conference*, 267-276.

[13] Henry, N., Bezerianos, A. and Fekete, J-D. (2008). Improving the Readability of Clustered Social Networks using Vertex Duplication. *IEEE Transactions on Visualization and Computer Graphics*, 14(6): 1317-1324.

[14] Henry, N., Fekete, J.-D. and McGuffin, M.J. (2007). NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309.

[15] Johnson, B. and Shneiderman, B. (1991). Treemaps: A space-filling approach to the visualization of hierarchical information structures, *Proc. 2nd International IEEE Visualization Conference*, 284-291.

[16] Kamps, T., Kleinz J. and Read, J. (1995). Constraint-Based Spring-Model Algorithm for Graph Layout, *Proc. Graph Drawing 95*, LNCS 1027, 349-360.

[17] Krempel, L. (1993). Simple representations of complex networks: strategies for visualizing network structure. *Presented at the 3rd European Conference for Network Analysis*, Munich.

[18] Noack, A. (2008). An Energy Model for Visual Graph Clustering. *IEEE Transactions on Visualization and Computer Graphics*, 14(6): 1333-1339.

[19] Shneiderman, B. (1992) Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92-99.

[20] Shneiderman, B., Aris, A. (2006). Network Visualization by Semantic Substrates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5): 733 - 740.

[21] Smith, M. A., Shneiderman, B., Milic-Frayling, N., Mendes Rodrigues, E., Barash, V., Dunne, C., Capone, T., Perer, A. and Gleave, E. (2009). Analyzing (social media) networks with NodeXL. *Proc. of the 4th International Conference on Communities and Technologies* (C&T '09), 255-264.

[22] Taylor, A. and Higham, D.J. (2009). CONTEST: a controllable test matrix toolbox for MATLAB. *ACM Transactions on Mathematical Software*, 35(4), Article 26.

[23] Wakita, K. and Tsurumi, Y. (2007). Finding community structure in mega-scale social networks, *Proc. 16th International Conference on World Wide Web*, 1275-1276.

[24] Watts, D.J. and Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440-442.