Linking Networks: A Study of Comparative Network Visualization

Leslie Watkins & Danielle Albers

CS 838-2 -- Visualization

ABSTRACT

The problem of visual network comparison is one that received relatively little exposure in the visualization literature. This paper describes a number of comparative network visualization approaches based on early explorations of small network comparisons. These prototypes have been applied to a set of flight path data between ten different cities for four different airlines as well as a larger synthetic data set. The ability of the visualizations generated by the study to depict relationships between the networks of the data sets suggests that the techniques of the study do provide a relevant comparison between different data sets. These techniques may later generalize to facilitate the comparison of a broader set of networks and contexts.

1 INTRODUCTION

Network visualization is one of the most well-known and well-studied form of information visualization. Generally, the purpose of these visualizations is to demonstrate the nature of relationships across a body of related data items. Graph theory provides a great deal of quantitative understanding of networks and a broad variety of literature exists with respect to understanding potentially massive data sets; however, little exploration has been done with respect to comparative network visualization: how to visual compare a set of distinct networks.

In this context, the notion of network comparison can be defined as a comparison of some subset of properties between different networks in a set. These properties can be observed at a global scale, as in layout properties, or a local scale, as in node or edge properties, or some combination thereof. The comparative network visualization must consider some subset of the network properties and display the similarities and discrepancies amongst each in a manner that is perceptually interpretable by the user.

The purpose of this exploration is to begin develop different methodologies for comparing network data. The scope of this issue is extremely broad with respect to the number of factors that must be considered when constructing such visualizations. In this study, we consider only a small set of labeled network with fewer than twenty nodes and unweighted edges.

These constraints serve to simplify the initial exploration of the concept. The purpose of this study is to begin to explore the realm of comparative network visualization. Ideally, by understanding how to compare small, simple network sets, we will have a better idea of how to construct visualizations that may scale in terms of both quantity and complexity. By understanding the problem at a small scale, we will be better equipped to address the issues at larger scales.

2 RELATED WORK

The study of network visualization has received a good deal of attention from the visualization community. However, a limited body of work exists addressing the notion of comparative network visualization. Erten et al. offer three different potential solutions to the network comparison problem using different weighting methods combined with weighted force-directed network layout in [2]. Both Collberg et al. [1] and Frishman [3] evaluate the use of time to express change over a series of networks. This mapping introduces time as a fourth dimension over which networks can be visualized and compared. Both of time-based methods suggest a sense of visualizing the evolution of a network over time.

A significant amount of work related to network comparison can be found in the domain of biology. A number of biological problems can be modeled as network diagrams whose comparison translates to an actual biological phenomenon. The uncertain mapping of biological networks to one another can reduce the problem of aligning two distinct biological networks into that of comparing two graphs of unlabeled node, essentially the graph isomorphism problem. Towfic et al. [6] developed a polynomial time solution to solving this alignment problem, which could be implemented in later iterations of this work in order to address the comparative network problem at a broader scope.

The existing work generally does not offer a viable solution to the problem of comparing multiple networks. We seek to use the lessons from visualization and existing network comparison methodologies in order to explore techniques for visualizing the comparison of multiple networks.

3 VISUALIZATIONS

In order to construct effective comparative network visualizations, it is important to consider the principles that illustrate successful comparison between networks. The user's *mental mapping* refers to the user's understanding of the underlying data. Network visualization must strive preserve the mental mapping of the component networks in order to form a visualization that is readily understandable by the user.

Furthermore, it is important to reduce the amount of visual clutter present in a given visualization. While a good deal of work

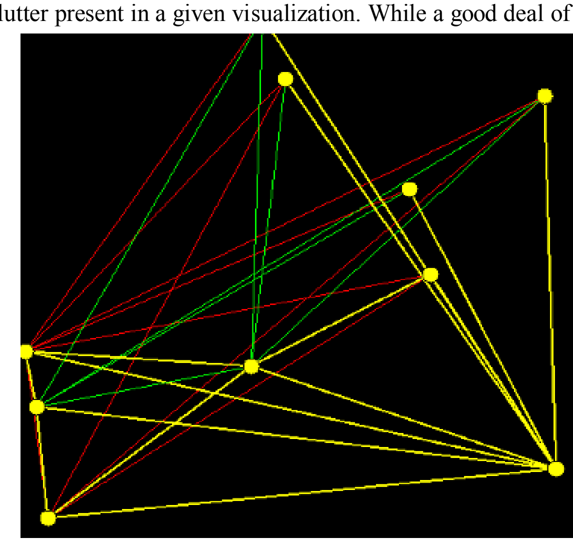

Figure 1. Blended Network for American (red) and United (green)

watkins@cs.wisc.edu, dalbers@cs.wisc.edu

has been done in this area with regard to standard network visualizations, it is nonetheless important to keep these lessons in mind with respect to comparative network visualization and keep the visual environment as simple and organized as possible.

The direction in which we approached this problem was through the development of a series of prototypes. The prototypes were tested over two data sets: a randomly generated set of nine different networks with variable levels of connectivity and a set of flight paths for ten major US cities serviced by Delta, United, American, and Continental Airlines. The format of the input data is an Excel file containing the connectivity matrix for the given input data set, with one matrix defined for each sheet in the workbook.

These matrices were then fed to the network prototypes in order to generate the comparative visualization. These prototypes are discussed in the following subsections.

3.1 Blended Networks

Blended networks use color blending to display two superimposed network diagrams. The nodes of the two graphs are laid out in a single layout (radial or force-directed) and plotted on the visual plane. Edges and nodes are then defined by their network belonging: those belonging to the first network are colored red, the second are green, and those belonging to both the first and second networks are colored yellow, the blend of the first and second color encodings.

This technique takes advantage of color to demonstrate similarity by using a literal blending of the two different graph encodings to signify shared elements. It preserves the overall network structures by drawing all the edges of both networks and maintaining edge position, even over common edges. It also minimizes visual clutter by maintaining a minimal number of objects in the visible plane. However, by coloring common edges with a different color that the rest of the networks, the user's sense of immediate belonging is disrupted and common edges may not immediately be perceived as part of each distinct network. Furthermore, this technique does not scale well beyond three networks, as color blending quickly becomes ambiguous.

3.2 Double-Edge Networks

Double-Edge Networks integrate the color blending and edge position ideas of Blended Networks while addressing the issue of network belonging. As in Blended Networks, the layout of the aggregated network (radial or force-directed) is computed and the nodes are laid out in visual space. Non-shared edges are laid out

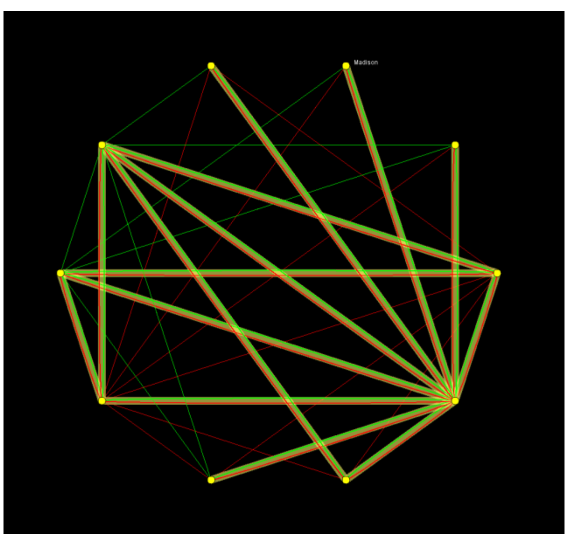

Figure 2. Radial Double-Edge Network for American Airlines (red) and United Airlines (green)

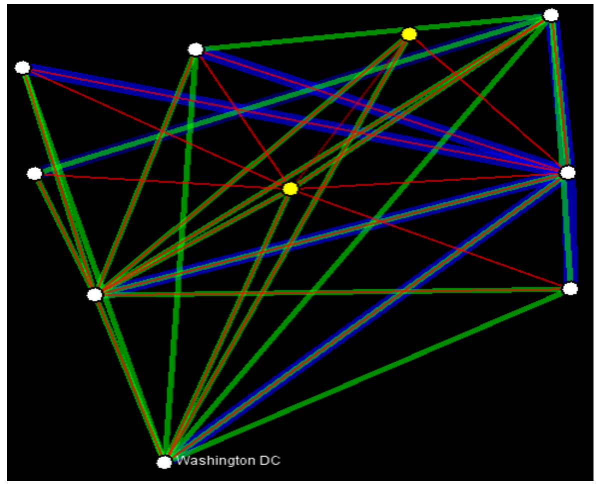

Figure 3. Overlay Network for American Airlines (red), United Airlines (green), and Continental Airlines (blue)

in the same manner as before. Shared edges are indicated via a pair of Bezier curves, one per network, bowing in opposite directions and bounded by a *ghosted* (opaque) yellow bounding box. The curves converge at the nodes at either end of the shared edge.

The use of the yellow bounding box preserves the Blended Network notion of color blending; however, using the ghosting technique allows the user to focus on either the shared edges or the individual network edges, thus maintaining the perceptual network sets. The convergence of the curves at the nodes preserves the mental mapping of the edges by always connecting node centers while the divergence keeps the edges visually distinct. This technique only extends to up to three networks (should edges common to all three networks be signified by a pair of curves with a center straight edge). Furthermore, the overlay of the different colors over shared edges can create simultaneous contrast artifacts and thus make it more difficult to trace the edge set belonging across certain networks.

3.3 Overlay Networks

Overlay Networks take advantage of visual prioritization to display the superimposition of multiple networks. As in the blended networks, the layout of the aggregated network (radial or force-directed) is first computed. The network edges are then laid atop one another, with the bottom-most network having the widest and most ghosted edges. Transparency and width are smoothly reduced across networks. Nodes are colored based on the color blending principle discussed in section 2.1.

This approach benefits from the fact that the original edge sets are visually preserved across all networks. The user can perceptually follow the distinct paths traced by the networks and the confusion of belonging found in the Blended Network is avoided. User focus can be dictated toward exploring individual graphs or any combination of comparisons. However, this technique does not well beyond three networks as the edges quickly grow too wide and begin to occlude visible space. Additionally, it relies on careful color selection to keep the network edges perceptually distinct.

3.4 Correspondence Network Matrices

Correspondence Network Matrices handle comparisons between larger numbers of networks by juxtaposing them in visual space. Visual space is broken down into a square matrix of *bounding boxes*, each of which hold a single network. When the user clicks

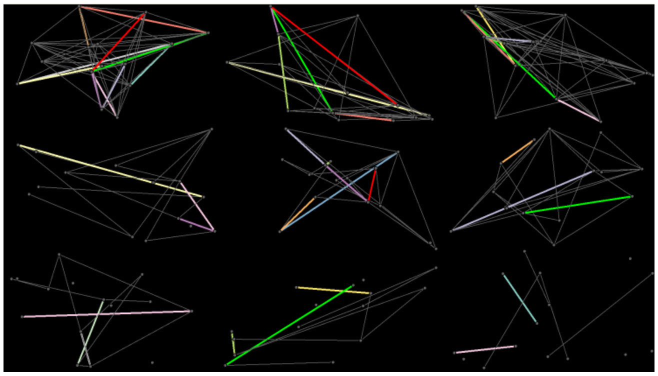

Figure 4. Correspondence Matrix Network for nine synthetic 15-node data sets

on an edge or node in a network, that edge or node is assigned a color over the entire network set. A twelve color ColorBrewer set defines the color library, and, as a result, at most twelve distinct nodes or edges may be selected at a given time.

This approach allows the user to compare networks in scenarios where the actual layout of the network is relevant. Furthermore, the technique could potentially scale to a relatively large number of networks (25 or more), bounded by the density of the networks and the size of the display. While the number of selected edges is limited, this limitation prevents confusion between nodes and edges which may otherwise be colored in a perceptually indistinguishable manner. This also insures that the occurrences of nodes and edges of interest across networks are readily recognizable.

However, this technique suffers from the fact that the degree with which networks can be readily compared is bounded by the proximity of a given set of networks. Networks further from one another are less readily comparable than those further apart. Additionally, densely clustered networks may become obscured as the size of a given network is reduced. Both of these limitations could be addressed through intelligent interaction and zooming techniques, but the application of such techniques is beyond the scope of this work.

3.5 Connections Application

This is a visualization of the difference between the adjacency matrices of two graphs. The edges that the two graphs have in common are solid, black lines. The edges unique to graph A are red, and the edges unique to graph B are blue. Colors were selected using ColorBrewer.

This visualization is designed to be interactive. The node placement is constant, and determined by a force-directed layout where all the edges of both matrices are considered. However, the user can highlight connections of only one matrix by clicking anywhere on the image to toggle between the two matrices (and right-clicking to clear). The connections are emphasized by the opacity of the lines.

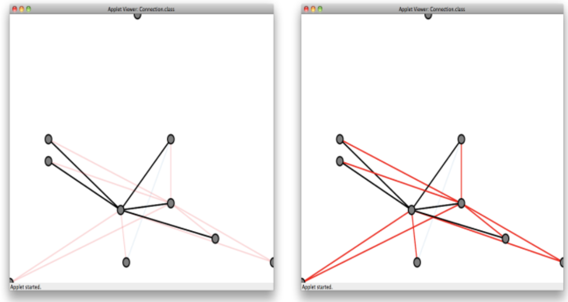

Figure 5. Flight information for Continental and Delta Airlines, with Delta highlighted in red.

3.6 Layout Application

Again, the “Layout” visualization shows the difference between the adjacency matrices of two graphs, and is interactive. The difference between this and the “Connections” tool is that, rather than emphasis being placed on the edges connecting the nodes, it is now on the placement of the nodes.

A force-directed layout is used, but now clicking on the image will change the edges that are considered when deciding node position. The locations of the vertices in the “cleared” image depend only on the connections that both matrices share. The user can toggle between a force directed layout that depends on the edges in graph A only and graph B only.

This tool may be used when the locations of the nodes are important. It does have the drawback of not preserving the user’s mental model very well, but the animated transitions do make up for this to some degree.

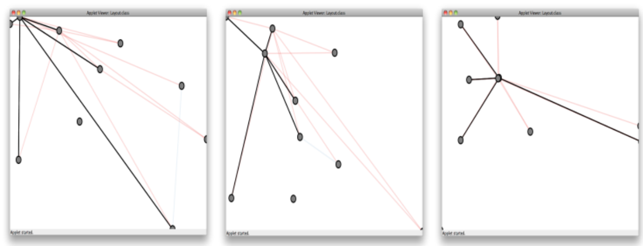

Figure 6. Flight information for Continental and Delta Airlines.

Both the Connections Application and Layout Application highlight the individual graphs while providing information about the other graphs for comparison. Although we only use two graphs in the examples, the number of graphs that can be compared this way is around five, limited by the color encodings.

5 PRE-PROCESSING

Our visualizations focus primarily on unweighted data with labelled nodes. This describes a small subset of all the information that can be visualized and compared using our techniques. With this in mind, we spent a portion of our efforts on studying the graph isomorphism problem, which can be generalized to graphs with weighted edges and unlabelled nodes.

Two graphs GA and GB are isomorphic if there exists a permutation matrix P such that their adjacency matrices satisfy the equation

$$PAP^T = B$$

If the two graphs are perfectly isomorphic, it implies that

$$|| PAP^T - B|| = 0$$

One way to find P is to test all possible permutations, and see if any satisfy the equality. This approach is extremely inefficient, and impractical for graphs with more than 10 nodes. However, it does always return the exact P. Another approach is to use eigenvalue decomposition to find a matrix that is similar, but not exactly equal to P.

Matrices A and B can be decomposed into a diagonal matrix of eigenvalues, and an orthogonal matrix of corresponding eigenvectors, given by

$$A = U_A \Lambda_A U_A^T$$
$$B = U_B \Lambda_B U_B^T$$

By substituting these decompositions into the permutation equality, we find that

$$P = U_B S U_A^T$$

where S is a diagonal matrix of entries 1 or -1. The matrix S is necessary because the eigenvectors of a matrix can be arbitrarily multiplied by any constant, and if they are multiplied by -1 they satisfy the eigenvalue decomposition equation given above. Supposing the value of S was known, then

$$\begin{aligned} tr(P^T \mathrm{U}_B S U_A^T) &= \sum_{i=1}^{n} \sum_{k=1}^{n} s_k\, h_{ik} g_{\pi(i)k} \\ &\leq \sum_{i=1}^{n} \sum_{k=1}^{n} |h_{ik}||g_{\pi(i)k}| \\ &= tr\left(P^T |\mathrm{U}_B||U_A^T|\right) \leq n \end{aligned}$$

due to the orthonormality of the eigenvector matrices. So $tr\left(P^T |\mathrm{U}_B||U_A^T|\right)$ is bounded above by *n*, and the inequality becomes an equality when an exact solution for P has been found. This means that an approximation to the graph isomorphism problem can be reduced to finding a P that maximizes $tr\left(P^T |\mathrm{U}_B||U_A^T|\right)$. This can be solved using the Hungarian algorithm, which runs in polynomial time.[8] Even when the graphs aren't perfectly isomorphic, a near optimal P (one that reduces $|| PAP^T - B||$) can be found using this method. The more nearly isomorphic the graphs are, the better this method performs.

We implemented both the brute force and eigendecomposition approaches to the graph matching problem. We found that the brute force method performs about 10% more accurately than the eigendecomposition problem, but is on the order of 1e6 times slower for a graph with 10 nodes.

6 DISCUSSION

Applying the above techniques to the airline data set reveals that these techniques can be used to draw conclusions about the overall relationship between the different networks. For instance, the Double-Edge Network in Figure 2 highlights the fact that American Airlines and United Airlines share one major hub and both also have a significant common presence in four other cities. The Overlay Network in Figure 3 reveals that every flight path in which Continental Airlines operates is also serviced by either American or United. The ability to draw such conclusions from these visualizations implies that the visual methods have succeeded in illustrating some useable of the networks.

Examining the different techniques, the use of color encoding plays a pivotal role in the display of patterns in the data. While this makes the information readily interpretable due to the mechanics of color processing in the perceptual system, this encoding can also prove to be a limiting factor to the eventual scalability of the techniques, as in the case of the Blended Network. One of the pivotal issues to developing large scale comparative network visualizations will be how to manage encodings with increasing larger amounts of available data.

Furthermore, the discussed techniques also operate over a set of small networks. As the size of the networks increases, the task of managing visual space in order to become clutter grows increasingly difficult. The mitigation of visual clutter is key to maintaining useful visualizations at scale.

In the future, understanding how to generalize the above techniques to a broader scope of problems, such as to unlabeled or weighted data, can lead to the establishment of more comprehensive and robust visualizations that facilitate a larger number of comparisons. Additionally, an analysis of the principles limiting the scalability of the different techniques explored above may provide insight into composing comparative network visualizations that scale to larger data sets.

7 CONCLUSION

The purpose of this exploration was to begin to explore the field of visual network comparison. In order to accomplish this task, a number of prototypes were constructed to visualize sets of small labeled networks.

The prototypes discussed in this paper succeeded in illustrating relationships between networks over both a synthetic and real world data set. This implies that the techniques developed here provide insight into how to successfully visualize network comparison.

Network visualizations are heavily used in a variety of different disciplines. Thus, the ability to compare between them offers valuable insight into a number of different domain problems. By developing a better understanding of the visual comparison of network visualizations, we simplify the task of using network visualizations across data domain and improve the overall usefulness of network visualization.

REFERENCES

[1] Collberg, C., Kobourov, S., Nagra, J., Pitts, J., and Wampler, K. 2003. A system for graph-based visualization of the evolution of software. In *Proceedings of the 2003 ACM Symposium on Software Visualization* (San Diego, California, June 11 - 13, 2003). SoftVis '03. ACM, New York, NY, 77-ff.

[2] Erten, C, Kobourov, S. G, Navabia, A. and Le, V. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *11th Symposium on Graph Drawing (GD)*, pages 437–449, 2003.

[3] Frishman, Yaniv and Ayellet Tal. Online Dynamic Graph Drawing, IEEE Transactions on Visualization and Computer Graphics, pp. 727-740, July/August, 2008.

[4] Ogata, H., Fujibuchi W., Goto, S., and Kanehisa, M. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Res*. 2000 Oct 15; 28(20):4021-8.

[5] Towfic, Fadi, Greenlee, M. Heather West, and Honavar, Vasant. Aligning Biomolecular Networks Using Modular Graph Kernels. *WABI*, 345-361, 2009.

[6] Frishman, Yaniv, and Tal, Ayellet. Dynamic Drawing of Clustered Graphs, Proc. *IEEE Symposium on Information Visualization*, 2004.

[7] Hoebe, Misja, and Bosma, Rien. Visualizing multiple network perspectives. *Dutch Directions in HCI* (Amsterdam, Netherlands, June 10, 2004)

[8] Umeyama, Shinji. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol.10 No. 5, 1998 Sept