CONNECTIONS 25(1): 37-47


# Visualizing Complexity in Networks: Seeing Both the Forest and the Trees

Cathleen McGrath
*Loyola Marymount University, USA*

David Krackhardt
*The Heinz School, Carnegie Mellon University, Pittsburgh, PA, USA*

Jim Blythe
*Information Sciences Institute, University of Southern California, USA*

*Visualization of complex relational information has become increasingly important as complex data and computational power have become more available to social network researchers. Common sources of relational complexity include change over time, multiple relationships, network size, and network density. The most useful method for displaying complex data often depends upon the source of complexity and the nature of the information to be learned. In this paper, we explore the use of motion, especially for representation of change over time and relationship. Also, using data from a large Wall Street investment bank, we demonstrate several strategies to represent complex relational data in two-dimensions.*

## INTRODUCTION

Information visualization has become increasingly important as access to large amounts of data increased and the ability to analyze that data has improved. More work is needed to refine methods of conveying the information uncovered back to users who must make sense of the vast amount of information that is available to them. Within the field of computer science we have seen an

increased interest in information visualization in general (Chen, 1999; Ware, 2000). And within the area of social network analysis, new approaches to the visualization of relational data have emerged such as Pajek (Batagelj and Mrvar, 1998) KrackPlot (Krackhardt et al., 1994) MultiNet (Richards and Seary, 1999). For a complete discussion of the history of visualization in social network analysis see Freeman (2000). The interests of these two communities are converging. Computer scientists are ever more cognizant of the importance of agent interaction. Social network analysts make use of the increasingly complex information and computational resources availableto study dynamic, multi-relational networks made up of many actors. Complexity is introduced as the number of actors and relations modeled increases and the interaction between network size, density and change over time describes complex systems. In terms of computation, as systems become more complex they require more time, space or resources to solve. Similarly complex systems require that users expend more effort for visualization. The proper design of visual displays of complex data will lessen the burden on viewers trying to make sense of data and on researchers trying to communicate important features of the displayed data.

The most well considered approach to visual representation of network data uses two-dimensional static displays. This approach tovisualization uses the spatial positioning of nodes; characteristics of nodes such as color, shape, and size; and characteristics of edges such as texture and color to communicate as much information as possible within a single graph. While we have learned a lot about the importance of layout and characteristics of nodes and edges for communicating information in graphs, we see that there are limits to the amount of information that can be displayedin a static graph.

Three major sources of network complexity are data that describe longitudinal and multi-relational networks and data that describe networks made up of many actors with dense connections. Each of these advances in our ability to collect and analyse social network data calls for advances in network visualization as well. In this paper we discuss approaches using KrackPlot to address complexity resulting from changes over time and among relationships and complexity resulting from large-scale networks. While some aspects of complex networks can be communicated using current two-dimensional graph representations, other aspects require new visualization techniques. For example, motion is very useful for conveying information about changes in networks, especially when the user needs to convey change over time or over network relationships. There are some cases where motion is not the best option for displaying complex information. The usefulness of motion may be limited by the size and density of the graph, the nature of the change that occurs and the media for publication of the information. We organize the rest of the discussion by the source of complexity in the network, and explore motion as well as techniques using two-dimensional representations for displaying complex network data.

## COMPLEXITY INTRODUCED BY NETWORK CHANGE

Motion is an obvious tool for displaying information about changes in networks over time, with the requisite warning that improperly applied, motion might obscure underlying phenomenon (Qin & Simon 1992a, 1992b and 1995). We havecompleted preliminary work (McGrath & Blythe, 2000) that suggests that viewers who are able to use motion to observe changes in graphs perceive those changes more quickly than those who are only able to switch between still representations of the

two time periods. Furthermore, we have shown that motion is particularly useful when used in conjunction with graph layout techniques that assign meaning to the positioning of nodes in Cartesian space (McGrath et al., 2001). In that case increased movement across representations highlights changes over time or across relationships. As expected, we found that motion is often useful for the efficient communication of network change. In response to these results, we have implemented a simple mechanism for displaying change across two graphs using motion in KrackPlot in order to allow users to employ motion easily.

***Displaying Change with Motion in KrackPlot using MORPHING***

The most recent version of KrackPlot includes a morphing function that allows users to take advantage of motion to display changes to the layout of a graph. The user first identifies an initial layout and then a second layout. Finally, the user designates the number of steps to be used to move from the first to the second layout. To use the morph function, users select "morph" from the top-level menu. Users may specify their first graph layout by selecting "first" while that layout is displayed on screen. Next the users can change the layout either by moving nodes and changing edges between nodes on the current graph interactively with the mouse or by loading a second version of the graph that contains the same nodes with different coordinates indicating their position on the screen. When the second layout is complete, the user selects "second" to identify the second layout. Now the "change" function will employ motion to display the change between the first and second layout.

The morphing function uses the shortest path between the first and second position of the each node in the graph. Any edges that change from the first to the second graph layout will change at the mid-point of the move from the first to second layout. The user can specify the number of steps between the first and second graph layout. The default number of steps between graphs is 50. If the number of steps is set to "1" the display flash between the first and second layout in "before and after" format. As the number of steps between the layouts increases, the motion appears to be more smooth. Currently, the morph function works best to convey changes in nodes' positions between the first and second layout. It does allow for adding and removing edges between the two layouts. It is more difficult to display changing nodes between the two layouts.

## COMPLEXITY INTRODUCED BY NETWORK SIZE

A second source of complexity results from network size and density. Large network are difficult to visualize because of the sheer size and amount of information available. Similarly, dense networks contain a large amount of relational data that is difficult to visualize. There are several strategies that users might take to visualize and represent large networks in a meaningful way. The best approach to visualizing large data sets often depends upon characteristics of the network (such as density) and the nature of the information to be learned from the graph. Three approaches that are often useful are (1) visualizing subsets of a large graph, (2) partitioning a large graph and visualizing relationships among partitions, and (3) exploring structure interactively using simulated annealing and adjusting the energy function to address characteristics of the displayed network. We will demonstrate these three approaches using data from Friedman and Krackhardt (1997). The data describe friendship ties among 83 support staff people in a large Wall Street investment bank. Shapes indicate ethnicity: ellipses denote Southeast Asian staff; rectangles denote Indian staff;

diamonds denote European staff; staff of all other ethnicity are represented with no shape around them. Figure 1 shows the friendship ties among bank staff in a circle layout. The circle does give an idea of the density of the graph however, it communicates very little about the structure of the network. Starting with this baseline method of representation, we will demonstrate the three techniques listed above.

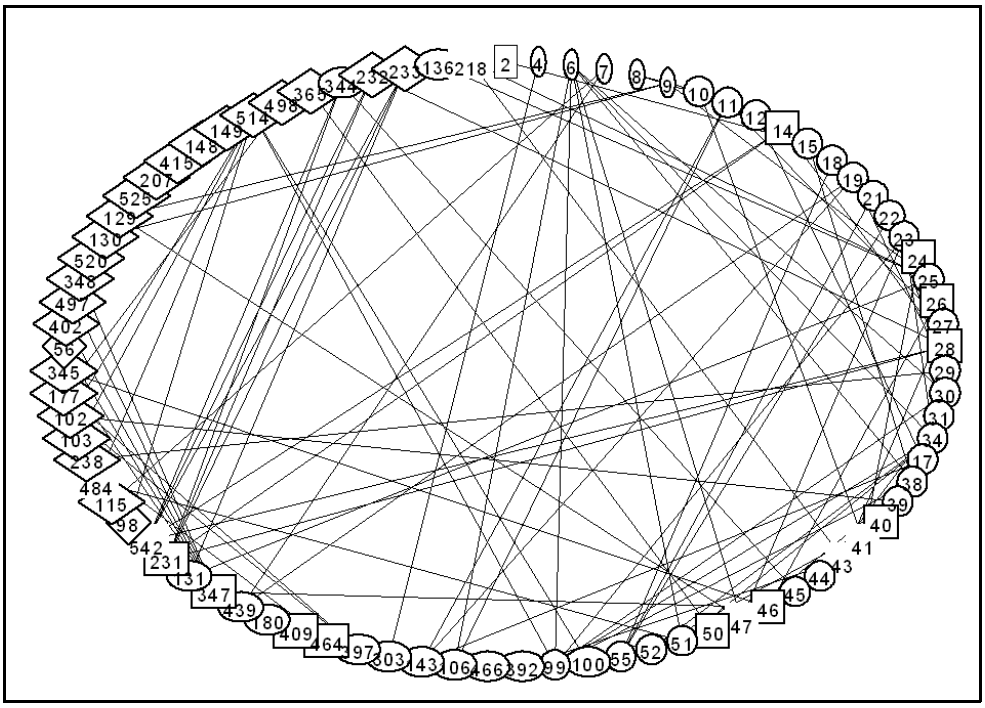


**Figure 1.** Friendship Network Circle Layout

***Visualize a subset***

In some cases, the user might find it helpful to break down the large graph into subsets. For instance, in the case of the investment banking data, the user might display the network for one group only. Here, we have extracted the European staff and Figure 2 shows that three groups of three or more people, two pairs of people, and seven isolates emerge when only European staff and their relations are considered. This approach limits the information that is displayed about the network (for instance, the actors who are displayed as isolates here might in fact be at the center of a group of non-European actors within the network). In some cases, it might be useful to compare images of the network even when they do not on their own tell the "whole story." The European network was displayed in KrackPlot by choosing the Hide option (Modify_ Hide_ Hide Type) to eliminate all other ethnic groups. Working interactively within KrackPlot, the user can hide and reinstate subgroups to explore the effect of their absence or presence on the overall structure of the graph. We could generate the same display for any subgroup or set of subgroups that have been identified as attributes in KrackPlot. A user might choose to represent a series of subsets of the graph to communication information about the graph.

***Partition, group, or block and display relationships between partitions***

The user may employ one of several techniques to identify roles or positions within a network. These relational approaches allow the user to translate complex networks into more simple forms.

Once nodes have been assigned to roles, the user can assign the same spatial positioning to all nodes in the same role by assigning the same Cartesian coordinates within KrackPlot. Figure 3 shows an

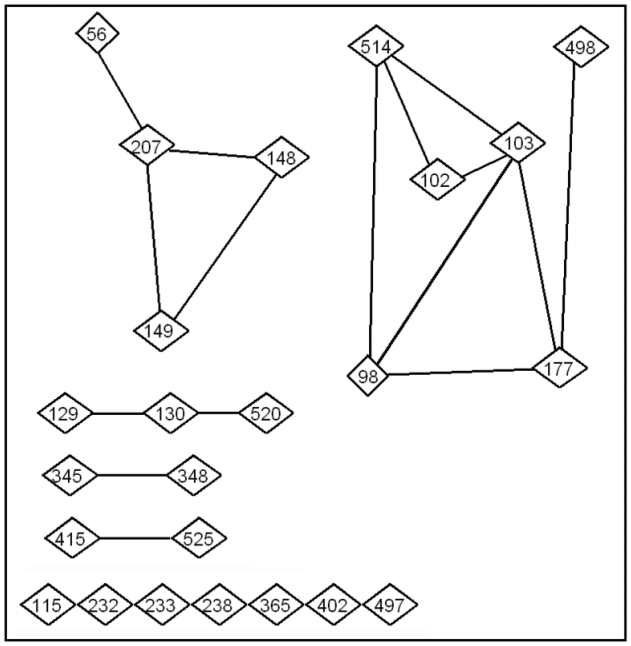


**Figure 2.** Friendship network for European

example of this using the same Wall Street investment bank data. We used UCINET V to identify roles using structural equivalence. Next we edited the KrackPlot file to assign the same coordinates to all nodes in the same role. We imported the edited KrackPlot file into KrackPlot. When the file is opened in KrackPlot, it appears as a reduced graph. In fact, all of the nodes are stacked up on each other. If the user holds the mouse over the stack and clicks, nodes will cycle through to the top of the stack. Any connections that exist among the structurally equivalent roles are portrayed as edges between the roles. Similarly, connections that exist within the structurally equivalent nodes in the same role are portrayed as self-reflected arrows back to the same position. To move a structurally equivalent group, the user must be careful to draw out a rectangle around the group. If the user just uses the mouse to drag the group, only the first node in the stack. We also added four nodes labeled "One" through "Four" to the representation to label the structurally equivalent roles. This approach pre-serves all of the information about the network while displaying only the reduced graph.

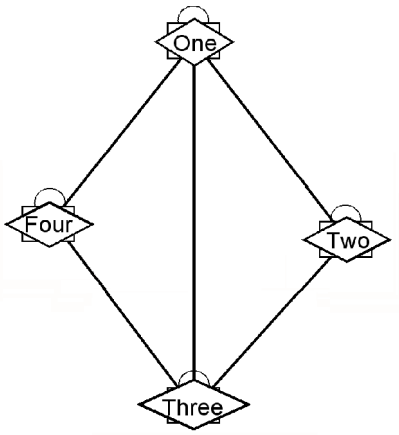


**Figure 3.** Reduced graph

***Use automatic layout to focus in on an overall structure and then substructures***

Users can explore large structures interactively within KrackPlot by using the available layout functions. Figure 4 shows network displayed using multi-dimensional scaling. While it is interest-

ing that the network looks like a velociraptor when displayed this way, it still conveys little information about the sttructure of the network. Figure 5 shows the network displayed using simulated annealing (and the default settings), an optimization routine that maximizes certain positive features of graph layout.

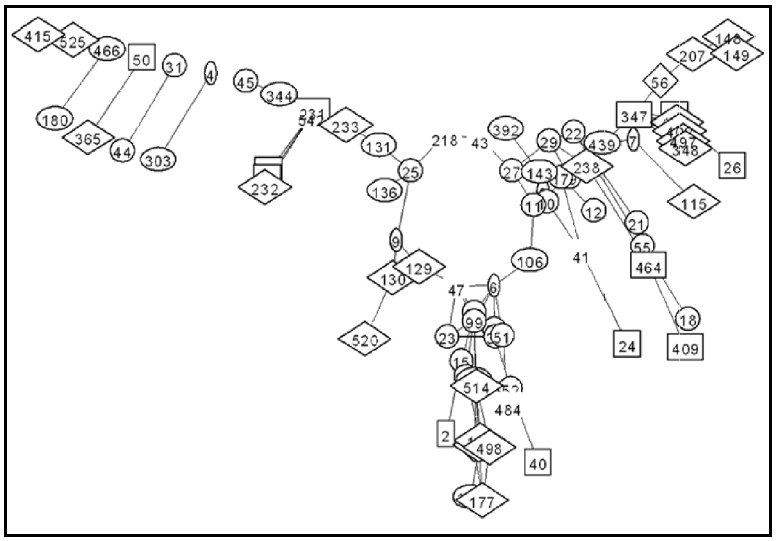


**Figure 4.** Multidimensional Scaling

Simulated annealing starts with the graph positions that are given. It then randomly moves an individual node to a point on the circumference of a circle defined around the node's current location, with a steadily decreasing radius. Once the node is moved, the routine reevaluates the new graph layout to determine if it is better or worse according to predefined graph layout features. The features are: nodes that are not too close to each other, edges that are not too long, edges that do not cross each other and nodes that do not go through edges. The "energy function", describing the attributes of the graph that will be optimized, is based on one suggested by Davidson and Harel

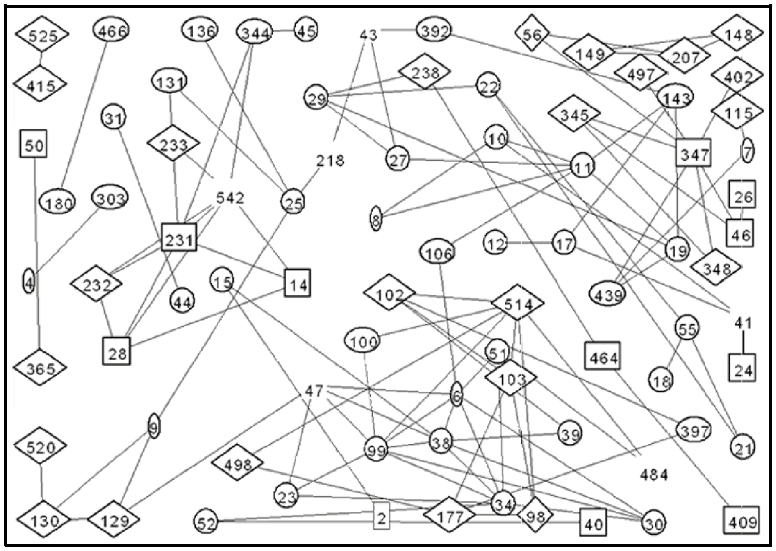


**Figure 5.** Anneal with default setting

(1989). If it is better, the new node position will be accepted. If it is worse, the new layout will be accepted with a small probability that depends on the "temperature" of the system and the amount worse that the new layout is. Simulated annealing will accept worse layouts with some probability to avoid local optima. Since simulated annealing begins the optimization routine from the initial graph layout, we have found it to be useful to start with a graph layout created using multidimensional scaling.

The default settings for annealing in KrackPlot were chosen for less complex graph. However, KrackPlot allows users to change the settings within anneal to attach different weights to each of the characteristics. The user can modify settings to accommodate more complex network information. In each annealing pass, the graph will be evaluated on these features using the default weights or the weights you provide in the "Settings" function. We will show how attaching different weights to the different graph features defined in the energy function can help the user uncover structure in a large graph. First, Figure 5 shows the bank staff friendship network layout using the default anneal settings. (Which are node repulsion = 1; edge repulsion = 0; edge repulsion in jiggle step = 1; edge length weight = 1; edge length variance = 0; edge crossing weight = 0; and weight on node hitting edge = 0.) The default settings for node repulsion keeps nodes away from each other and the default setting for edge length keeps connected nodes close to each other by keeping the edges between them short. As seen in Figure 5, using the default settings for simulated annealing does not uncover much structure in an 83-person network. When there are many nodes, the default setting for node repulsion tends to force the nodes to spread out too evenly on the page. The energy function pushes against the outside boundary of the picture.

Users can uncover more information about the structure of large graphs by working interactively with the annealing layout function and changing the weight attached to the annealing energy function. Also, using the “jiggle” function allows users to make refinement to the previously annealed layout.

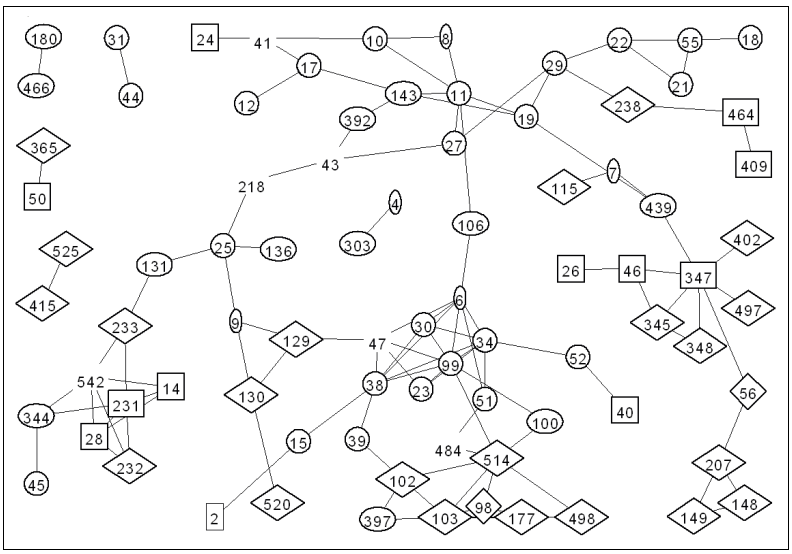


**Figure 6.** Anneal with Edge crossings=0 and Node repulsion=.1

The first change that we employ is to the weight assigned for node repulsion. Since there are 83 nodes in the network, we lower the node repulsion weight so that the nodes do not simply push against each other and against the boundary of the picture. We hold the edge length weight

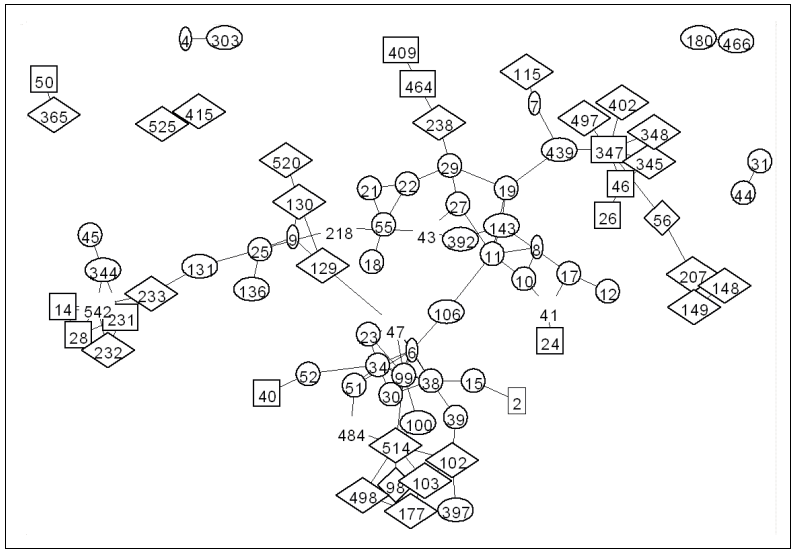


**Figure 7.** Anneal with Edge crossings=0, Node Repulsion=.01

constant at 1 so that in all versions of the layout, connected nodes stay close to each other. Figure 6 shows the result of lowering the node repulsion weight from 1 to .1; now the structure of the network begins to be more visible.

As we lower the node repulsion to .01 in Figure 7, we see that some groups and cluster emerge from the network. Figure 8 shows how the layout can be refined using the "jiggle" function in KrackPlot.

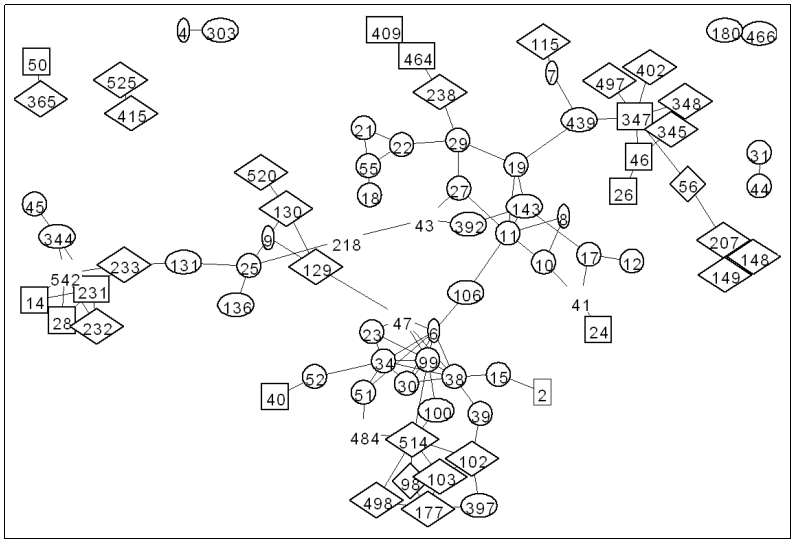


**Figure 8.** Jiggle prior picture with Edge Crossing for jiggle=.01

At this point, we introduce attention to limiting edge crossing by raising the weight on edge crossings from 0 to .01. Figure 8 shows little change from Figure 7, but the jiggle function does attempt to refine the display by reducing edge crossings. Figure 9 shows the result of using simulated annealing with the node repulsion set even lower, to .001. Here the connected nodes form into tight

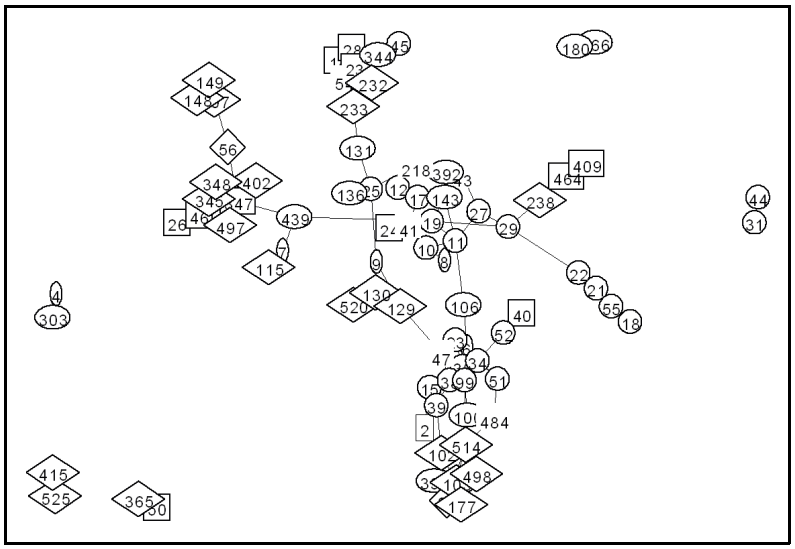

**Figure 9.** Anneal with edge crossings=0 and node repulsion=.001

clusters of overlapping nodes. While on its own, this display makes it rather difficult for viewers to extract information, it provides a good starting point for the jiggle refinement that is shown in Figure 10. For the layout displayed in Figure 10, we have taken the prior layout and refined it by including an edge crossing weight of .01 and increasing the node repulsion weight to .01. This

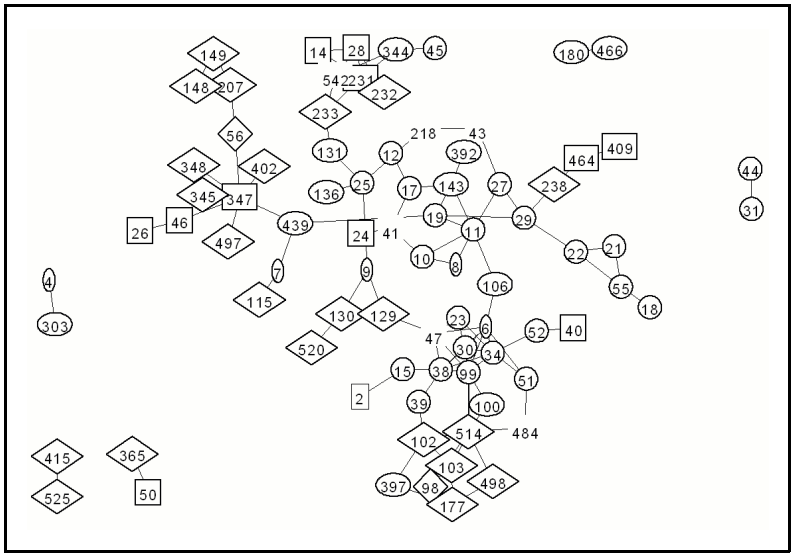

**Figure 10.** Jiggle prior picture with edge crossings=.01 and node repulsion=.01

process leads to a readable graph in which the structure of the network of 83 people emerges from the representation.

## CONCLUSION

In this paper, we demonstrated several techniques for displaying complex network data using current software. We recognize that no one technique is best for displaying structural data under all circumstances. The amount of information, its complexity, and the salient features that are to be discerned in the picture all affect the choice of methods for display of such data. For example, in some cases, motion will be helpful; in other cases motion will obfuscate. Often times, as in the last investment banking example, there is a tradeoff between seeing the overall forest – the clusters of overall groups and their relative social proximity or ordering in relation to each other – and seeing the finer detail of the trees – identifying key players and roles within these groups. Interacting back and forth, changing the parameters to identify macro groups and then relaxing these parameters to reveal finer detail, provides a model for future work in automated and interactive display of complex network data of this kind.

But all of this progress leads us to ask another question. While we can produce faster and fancier network display programs, it is important to keep in mind that the purpose of all of this is to communicate important features of the structure to the viewer. To fully answer the question "What display technique is best" or even "What display technique is best in this circumstance", we need to better understand what the perceiver is learning or seeing in the display. To be sure, we can make more programs that seem to us as researchers/ programmers to make "better" pictures; but we are relatively ignorant of how general human perception interacts with these new fancy features. We believe that both kinds of research should go on in parallel. That is, new techniques of display should be evaluated for their ability to communicate structure to the user, and in turn new techniques can be developed to take advantage of what we find people "see" in particular display formats. Currently, it is our impression that there is a severe imbalance in this parallel research: Much more emphasis has been placed on methods of network data display and little on what people infer from structural displays. We certainly applaud the rapid progress in technology and would not want to diminish the enthusiasm of those who are diligently working to make finer and finer programs. We simply suggest that the other side of this progress also deserves equally enthusiastic efforts to reveal the human gain in understanding of social structures.

## REFERENCES


Chen, C. 1999. *Information Visualisation and Virtual Environments*. London: Springer-Verlag.

Batagelj, V. and A. Mrvar. 1998. Pajek - Program for large network analysis. *Connections* 21: 47-57.

Friedman, R. and D. Krackhardt. 1997. Social Capital and Career Mobility. *Journal of Applied Behavioral Science*. 33: 316-334.

Krackhardt, D., J. Blythe, and C. McGrath. 1994. KrackPlot 3.0 An Improved Network Drawing Program." *Connections* 17: 53-55.

Freeman, LC. 2000. Visualizing Social Networks. *Journal of Social Structure*. 1(1).

McGrath, C. and J. Blythe. 2000. Influence of motion on human perception of dynamic networks. Presented to the XX International Social Network Conference. Vancouver, British Columbia. 2000.

McGrath, C., J. Blythe, and D. Krackhardt. 2001. Understanding human perception of dynamic representations of social networks. Presented to Schloss Dagstuhl Seminar No. 01271 on Link Analysis and Visualization. Saarbrucken, Germany.

Qin, Y., and H.A. Simon. 1992a. Imagery and mental models in problem solving. *American Association for Artificial Intelligence Spring Symposium Series: Working Notes*, 18-23.

Qin, Y., and H.A. Simon. 1992b. Imagery as process representation in problem solving. In *Proceedings of the 14th Annual Conference of the Cognitive Science Society*, 1050-1055.

Qin, Y., and H.A. Simon. 1995. Imagery and mental models in problem solving. In *Diagrammatic reasoning: computational and cognitive perspectives*, J. Glasgow, N.H. Narayanan, and B. Chandrasekaran, Eds.,403-434. Menlo Park, CA: AAAI/The MIT Press.

Richards, W. and A. Seary. 1999. MultiNet. Software tool.

Ware, C. 2000. *Information visualization: Perception for Design*. Morgan Kaufmann Publishers.