Visualizing very large layered graphs with quilts

Benjamin Watson, David Brink, Matthias Stallman, Ravi Devajaran, Theresa-Marie Rhyne and Himesh Patel

Abstract—Traditional node-link depictions of layered graphs such as flow charts and process or genealogy diagrams are in widespread use. Layers emerge from applied context (e.g. process stages or familial generations), or are inserted to improve visual clarity. However for many applications these diagrams quickly lose their utility as graph complexity grows. Layout algorithms such as crossing minimizers can preserve utility for larger graphs, but also quickly reach their limits. We introduce quilting, an interactive, matrix-based depiction for very large layered graphs that remains useful even when optimized node and link depictions have become unintelligible. We demonstrate quilting using an activity-based management (ABM) application that must depict layered graphs with thousands or even hundreds of thousands of nodes. Unlike node-link depictions, quilts depict 500-node graphs quite clearly. On typical desktop displays, quilts depicting larger graphs must be summarized.

Index Terms—graph drawing, layered graphs, crossing minimization, matrix depiction

1 QUILTING FOR DEPICTING LAYERED GRAPHS

Layered graphs such as structure charts, process diagrams, and flow charts have wide-ranging application. In these graphs, nodes are grouped into layers defined either by the application context, or introduced to increase visual clarity.

Traditional, node-link depictions of layered graphs arrange members of a layer into a line [3][13]. *Proper links* connect nodes on adjacent layers; we call remaining links *skip links*. Crossing minimization algorithms reduce the intersection of proper links, and can improve legibility. Nevertheless, as the number of links grows, these depictions can become quite muddled, with viewers having trouble understanding graph connectivity (Figure 3).

Scalability of depictions to complex graphs is also a problem for large, unlayered graphs. Matrix depictions, first suggested by Bertin [4], offer a good solution. Ghoniem et al. [11] compared the usability of node-link and matrix depictions of these graphs and found that for graphs with more than 20 nodes, matrix depictions were much clearer, with the exception of path-finding in moderately sized graphs.

To address the scalability problem for layered graphs, we introduce *quilts* (Figure 1), a new depiction that uses matrices to visualize layered graphs. Below, we review existing methods for depicting layered graphs, describe the meaning and manufacture of quilts, and show how quilts might be used in one application.

2 DEPICTING LAYERED GRAPHS

There are many layout algorithms for layered graph depiction, the best known of which is the STT method [17], based on prior work by Warfield [18] and Carpano [5]. Most methods for depicted layered graphs have three phases: layer assignment, crossing minimization, and horizontal placement. In this section, we sketch the approaches taken in these phases by this and other algorithms [3][9].

If layers are not derived directly from the application, a layout algorithm [7][8][14] can introduce them by minimizing one or more of the following objectives: (a) height, that is the number of layers (b) height given a fixed width, and (c) dummy nodes, that is the total number of layers skipped by skip links. Note that minimizing the width by itself is trivial – assign one node to each layer – but does not lead to aesthetically pleasing drawings.

To minimize link crossings, algorithms permute the nodes on each layer. While this is an NP-hard problem with even just two layers [6][10], several good heuristics are available, most based on sweeping from top layer to bottom, then from bottom to top, iterating a predetermined number of times [3][9]. During each sweep, an algorithm fixes the order in the current layer while permuting the next, then fixes the order of the newly permuted layer order as a starting point for the layer after that.

The final phase positions each layer at particular y-coordinates, attempting to minimize the number of bends in the skip links and/or the total/max deviation from vertical of the links issuing from the layer. This is a quadratic programming problem.

As a final note, the graphs in layered depictions need not be acyclic nor even directed. If there are directed cycles, a layout algorithm can temporarily reverse some link directions to create an acyclic graph. Minimizing the number of such links is NP-hard but a reasonably good greedy algorithm exists [3]. If the graph is undirected there are many ways to assign directions to the links so as to create a directed acyclic graph (DAG) – any numbering of nodes defines a DAG if we direct all links from lower to higher numbered nodes. The main problem with this numbering is that it may not be clear what sort of numbering will result in an aesthetically pleasing depiction.

3 QUILTING LAYERED GRAPHS

The quilt depiction of a layered graph is a simple adaptation of the matrix depiction for unlayered graphs [4]. We represent proper links with an achromatic matrix, and chain these matrices together with additional colored *levels*: rows or columns of cells representing layers (Figure 1). Each level cell corresponds to an individual layer node. To distinguish levels from one another and from matrices, we assign a unique chroma to each level. We assign each level cell a unique saturation, effectively making the color of every graph node unique. Small dots indicate the presence of proper links in matrices.

Each level acts as the *source level* for the following matrix, and the *destination level* for the previous matrix. The *first level* contains nodes without incoming proper links, and is therefore only a source level. The *last level* contains nodes with no outgoing proper links, and therefore is only a destination level. We number the l levels from first to last, starting with 1. Level numbers in the quilt increase as one moves from left to right, and top to bottom.

Nodes on odd-numbered levels are lined up horizontally while those on even-numbered levels appear vertically. A simple link from an odd to even level makes a left turn whose corner is the cell

- *Benjamin Watson is with NC State University, Email: bwatson@ncsu.edu.*
- *David Brink is with the SAS Institute, Email: david.brink@sas.com.*
- *Matthias Stallman is with NC State University, Email: matt_stallman@ncsu.edu.*
- *Ravi Devarajan is with the SAS Institute, Email: ravi.devarjan@sas.com*
- *Theresa-Marie Rhyne is with NC State University, Email: tmrhyne@ncsu.edu.*
- *Himesh Patel is with the SAS Institute, Email: himesh.patel@sas.com*

representing that link. Similarly, a simple link from an even to odd level makes a right turn through the cell for that link.

We depict skip links with colored cells at the end of the matrix row or column that follows level nodes. The color of each skip link is the same as its destination level node (recall that the color of each node is unique). To improve clarity, we sort the skip links using level number as the primary key and cell number (within the level) as secondary key.

3.1 Interactivity

To ensure good scalability, quilt cells are typically too small to support display of application-assigned node properties, especially textual labels. Viewers can reveal a node or link's properties by hovering over its corresponding cell (Figure 2).

As noted by Ghoniem et al. [11], one of the weaknesses of a matrix depiction is path following. We address this shortcoming by allowing viewers to highlight graph paths in the quilt by clicking on node or link cells (Figures 1 and 2). The first such click highlights the cell itself, as well as immediately adjacent nodes. Each additional click highlights nodes that are one more link distant. We call this *click-through*. A backward click-through, removing highlights from the nodes last reached, is available by clicking on the node while the control key is pressed. Pressing the shift key while clicking on a node highlights all the nodes reachable from the clicked cell. Clicking on any portion of the depiction not containing a node or link removes the highlight.

To provide viewers with the strengths of both quilts and the node-link depiction, we couple these depictions in an interactively linked view (Figure 2). Clicking in either depiction will highlight the appropriate graph path in the other depiction.

3.2 Summarization

When graphs contain thousands of nodes or more, they can require summarization, since the corresponding quilt depiction will not fit in a typical display. To summarize quilts, we adopt the methods of much prior work [1][2][6][16]: we cluster nodes (Figure 5). In this section we describe how we create a simple clustering that ensures that the entire quilt fits in the display, and how we adjust the quilt to better depict summarized graphs. In future work, we will investigate more optimal and application-appropriate clusterings, as well as an effective zooming depiction.

Each summarized level cell except the last represents s unsummarized nodes, with the first summarized node representing unsummarized nodes 1 through s, the second the unsummarized nodes s+1 through 2s, and so on. The last summarized node represents n_i mod s nodes, where n_i is the number of nodes on the level i, $1 \leq i \leq l$.

Each summarized matrix cell represents up to s x s links. In applied settings, we find that summarized matrix cells represent far fewer links. We map the number of links in summarized matrix cells to cell luminance, with darker cells indicating more links.

Each summarized skip link cell represents up to s x s links from the same summarized source level cell (depicting up to s nodes) to the same summarized destination cell (depicting up to s nodes). As before we indicate the level of the summarized destination cell with chroma, and the destination cell's position within that level with saturation.

We make s as small as possible, i.e., summarize as little as possible, while still allowing the quilt to fit into the display. If screen height and width in pixels is S_w and S_h, and quilt height and width is Q_w and Q_h, $s = \max(\text{ceiling}(Q_w/S_w),\text{ceiling}(Q_h/S_h))$.

Hovering over a summarized level cell indicates the number of cells being summarized, as well as statistical summarizations of any application properties in the summarized set of cells. For example, if one property indicates the one of a few classes to which a node belongs, hovering will indicate not only the number of summarized nodes, but also the number of nodes belonging to each class. Hovering over summarized link and skip cells has similar functionality.

Selecting a summarized cell highlights summarized level and matrix cells in a manner much like that in unsummarized quilts.

We have not yet implemented a zooming function, though that is one of the first on our list of improvements (see below).

3.3 Strengths, limitations and issues

Scalability. Quilts remain legible even as graphs grow to contain thousands of nodes. In contrast, node-link depictions with only a few dozen nodes can be difficult to understand.

Property visibility. Quilt scalability comes at the price of some property visibility, with node and link properties and labels hidden inside small cells until the user hovers over the cell. This can make finding nodes or links difficult. However, property visibility in node-link depictions also suffers as graphs grow in size. Although we have not implemented this functionality, it would be trivial to use larger, labeled quilt cells when graphs are small.

Link visibility. In order to determine if two nodes in a quilt are linked, one must search for the matching link cell by tracing the matching row and cell across the matrix that joins them, or by examining the color of the cell's skip links. In node-link depictions this can be much simpler, as all link depictions begin and end directly at the nodes concerned. Nevertheless in large graphs, it quickly becomes difficult or impossible to follow links in node-link depictions.

Skip links. Quilt legibility suffers as the ratio of skip to proper links grows, increasing graph complexity (Figure 4). When the number of skip links is large, layers become a less useful visual grouping. Quilt depictions of such graphs will have fewer dots in their quilt matrices, wasting more display space; and more cells outside these matrices, where there is less visual structure. Node-link depictions of the same graphs contain more crossings and meandering links that are difficult to remove through optimization. Perhaps the most effective depiction of such irregularly connected graphs is a single matrix, which does not require special external cells to represent skip links, because it does not depict layers.

Crossing minimization. In node-link depictions, preserving legibility as graph size moves from small to moderate requires crossing minimization – an NP-hard problem. Quilt depictions remain legible even without crossing minimization, though such minimization may still prove beneficial (see our Future Work section).

3.4 An applied example

We developed the quilt in the applied context of SAS's Activity-Based Management application (ABM). ABM is an analytical application that models an organization's processes to determine accurately the cost and profitability of products and customers. ABM uses the same "numbers" found in an accountant's general ledger, but instead of viewing cost and revenue centers in traditional hierarchies organized by group, division, etc. or by products or services, ABM models the interactions between the groups and assigns revenue to those responsible for the products or services. This allows the true nature of cost and revenue to be determined.

SAS's ABM system uses a directed graph to model the interactions within an organization. Large organizations and their processes yield complex models often containing hundreds of thousands of vertices and millions of links. While it is easy to answer specific cost or revenue questions in complex models, the true wealth of this information must be discovered by visualizing the model to understand hidden trends. Do the processes for one product correlate to another? Do profitable products have more efficient processes? Are the most profitable customers linked to the most profitable products? Unfortunately, most of our ABM graphs overwhelm traditional node-link depictions, making visualization – the most intuitive method of analysis and query – inefficient at best and often futile.

Figure 2 shows a small ABM graph depicted using linked node-link and quilt views. The graph models the cost processes involved in an airlines' flight catering service. Here we have highlighted the food prep group in our linked views. By following the incoming links we see that the objects that contribute to this group are kitchen salaries, kitchen utilities, kitchen appliances, install computers, maintain computers, review resumes, and maintain healthcare programs. By following the outgoing links we see that the group contributes to assemble/load main trays, schedule order delivery, store items, inspect items, wash food items, assemble/load hot trays, order supplies, clean work area, sanitize work area, and deliver/pickup containers. Each set of these incoming and outgoing objects have their own respective cost associations, which affects the cost efficiency of the food prep group up and down its chain of dependencies. Next to the food prep group is the cook group. By selecting it (Figure 2 bottom), we find that even though the cook group has the same set of incoming links, its outgoing links differ from the food prep group except for ordering supplies and clean work area. Cooks are responsible for ordering supplies, cutting food items, preparing mixtures, cleaning work areas, and cooking/baking food items.

This example illustrates some of the strengths and weaknesses of quilts. Even with highlighting, in the node-link depiction it is difficult to discern exactly how highlighted nodes are connected to the food prep group; in the quilt this connectivity is much clearer. On the other hand, in the quilt we cannot see the labels or attributes of any nodes except the one selected.

4 Conclusion and Future work

We have presented a new depiction of layered graphs, which we believe remains legible even when the depicted graphs have several hundred nodes. Our work on this depiction has only just begun. In particular we plan to investigate:

Layer and crossing optimization. The application of these optimizations for node-link depictions to quilts will minimize skip links, and reorder level cells to group the dots representing proper links around the diagonal. We plan to study the effect of these optimizations on quilt clarity.

Clustering and zooming in summarization. Alternative, application based clusterings of the nodes on the graph may prove useful. These clusterings will have to be formed within level, to keep quilt levels distinct. Our first zooming implementation will add an inset detail view. A possible alternative would embed the detail view into the summarized quilt. This will be more challenging, requiring layout adjustments to the neighboring levels and matrices.

Interactive editing. interactive user adjustment of node-link depictions of layered graphs [15] is a difficult problem, given the complex optimizations they require. Interactive editing of matrix depictions of unlayered graphs [12] does not introduce the same layout challenges. Editing quilt depictions of layered graphs might offer similar advantages. One new wrinkle would be moving a node between levels. Also, ordering of rows and columns in matrix representations of unlayered graphs according to two or more criteria is difficult [11][12][19]. The new level structure in quilts may offer a potential solution.

Relation to OLAP. Our ABM application also makes heavy use of OLAP methods. We plan to explore the possible relationship of quilts and Wattenberg's PivotGraphs [19].

User interface improvements. To improve link legibility, we plan to implement backward click-through of link highlights, removing highlights link by link. To improve legibility of node and link properties, we may implement a hover radius, so that the properties of any nodes or links within that radius of the cursor are displayed. This should make locating nodes and links much easier.

User study. It would be extremely useful to study the effectiveness of quilts in comparison to node-link depictions of layered graphs, in much the same way that Ghoniem et al. [11] compared node-link and matrix depictions of unlayered graphs.

Acknowledgements

Our thanks to the Visualization group at SAS, as well as the Center for Visualization and Analytics at NC State, who nuture the NC State/SAS collaboration. Scott Sutton of the Activity Based Management solution group at SAS also has our gratitude for his help in developing quilts for SAS's ABM application, and for his improvements to this manuscript. Thanks are also due to Frank Wimmer of the UI Analysts group at SAS, as well as Paul Layne in the graph development group for their timely help in the implementation of the test application.

5 References

[1] J. Abello and J. Korn. 2002. Mgv: a System for Visualizing Massive Multidigraphs, IEEE Transactions on Visualization and Computer Graphics,vol. 8 (1), pp 21–38, 2002.

[2] J. Abello and F. van Ham. 2004. Matrix Zoom: A Visual Interface to Semi-External Graphs, Proc. IEEE Symp. Information Visualization 2004, IEEE CS Press, pp 183-190, 2004.

[3] Battista, G.D., Eades, P., Tamassia, R. and Tollis, I.G. 1999. *Graph Drawing: Algorithms for the Visualisation of Graphs*, PrenticeHall.

[4] Bertin, J. 1967. Sémiologie graphique : Les diagrammes - Les réseaux – Les cartes. Editions de l'Ecole des Hautes Etudes en Sciences, Paris, France.

[5] Carpano, M.J. 1980. Automatic display of hierarchical graphs for computer aided decision analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, *10*, 705-715.

[6] P. Eades, and Q.W. Feng. 1996. Multilevel Visualization of Clustered Graphs, Proceedings of the 4th Intl. Symp. On Graph Drawing, LCNS 1190,Springer-Verlag, pp 101-112.

[7] Eiglsperger, M., Siebenhaller, M., and Kaufmann, M. 2005. An efficient implementation of Sugiyama's algorithm for layered graph drawing. *Journal of Graph Algorithms and Applications*, *9*, 305-325.

[8] Gansner, E.R., Koutsofios, E., North, S.C., and Vo, K.-P. 1993. A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, *19*, 214-230.

[9] Gansner, E.R. and North, S.C. 2000. An open graph visualization system and its applications to software engineering. *Software - Practice and Experience*, *30* 11. 1203-1233.

[10] Garey, M.R. and Johnson, D.S. 1983. Crossing number is NP-complete. *SIAM Journal of Algebraic and Discrete Methods*, *4,* 312-316.

[11] Ghoniem, M., J.-D. Fekete, P. Castagliola. 2004. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. IEEE Symposium on Information Visualization.

[12] Henry, N. and Fekete, J.-D. 2006. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions On Visualization And Computer Graphics, 12*, 5.

[13] Herman, I., G. Melancon, M. Scott Marshall. 2003. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, *6*, 24-43.

[14] Nikolov, N.S, Tarassov, A., and Branke, J. 2005. In search for efficient heuristics for minimum-width graph layering with consideration of dummy nodes. *ACM Journal of Experimental Algorithmics*, *10,* 1-27.

[15] K. Ryall, J. Marks, S. Shieber. 1997. An Interactive Constraint-Based System for Drawing Graphs. Proc. ACM UIST, 97-104.

[16] D. Schaffer et al., 1996. Navigating hierarchically clustered networks through fisheye and full-zoom methods, ACM Transactions on Computer-Human Interaction, vol 3(2), pp 162-188.

[17] Sugiyama, K., Tagawa, S., and Toda, M. 1981. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, *11*, 109-125.

[18] Warfield, J.N. 1977. Crossing theory and hierarchy mapping. *IEEE Transactions on Systems, Man, and Cybernetics*, *7*, 502-523.

[19] Wattenberg, M. 2006. Visual Exploration of Multivariate Graphs. Proc. ACM CHI, 811 – 819.

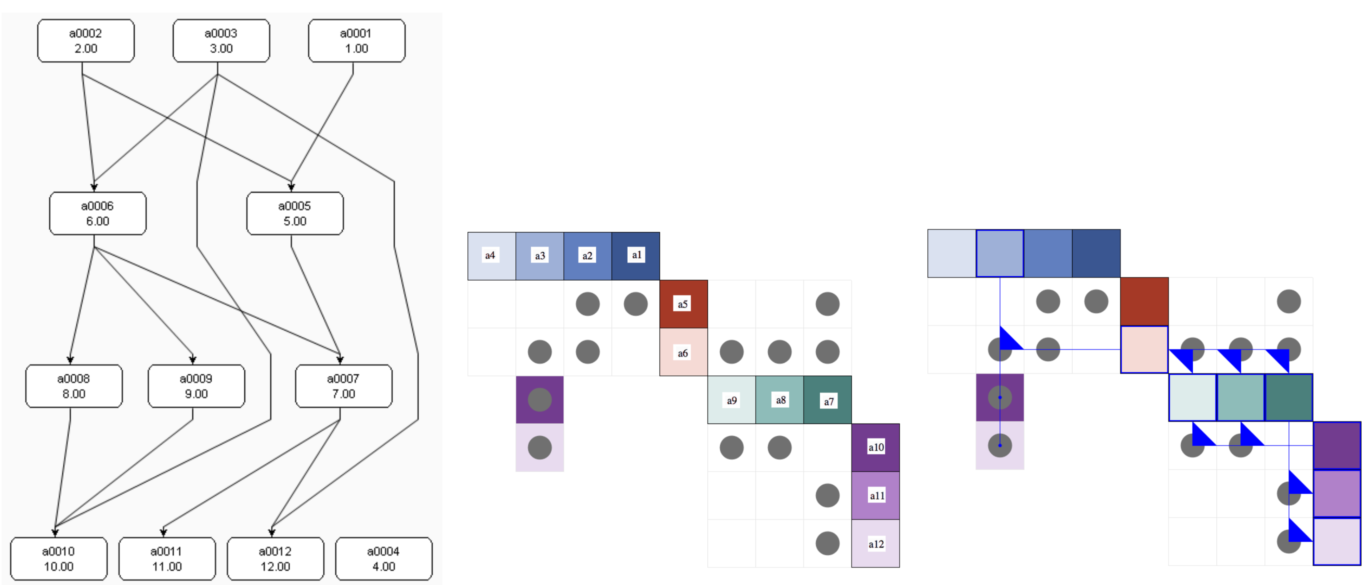

Figure 1: a simple graph. A node-link depiction on the left. Center, a quilt depiction, annotated in this introductory figure. Right, one of the nodes in the first level is selected, highlighting the links emanating from it.

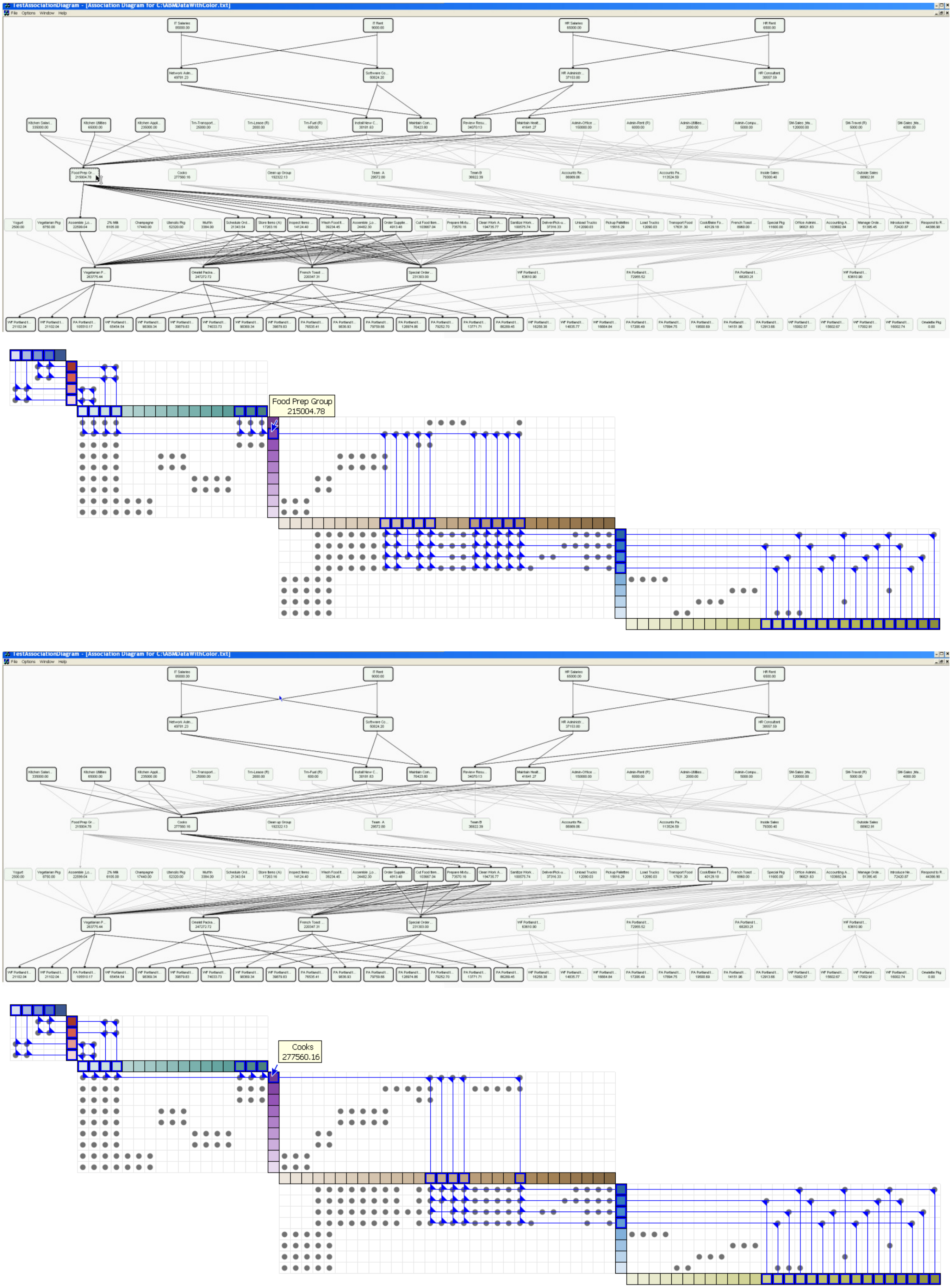

Figure 2: A small applied example from SAS. In the top two, the user has selected the “food prep group” node; in the lower two, “cooks”.

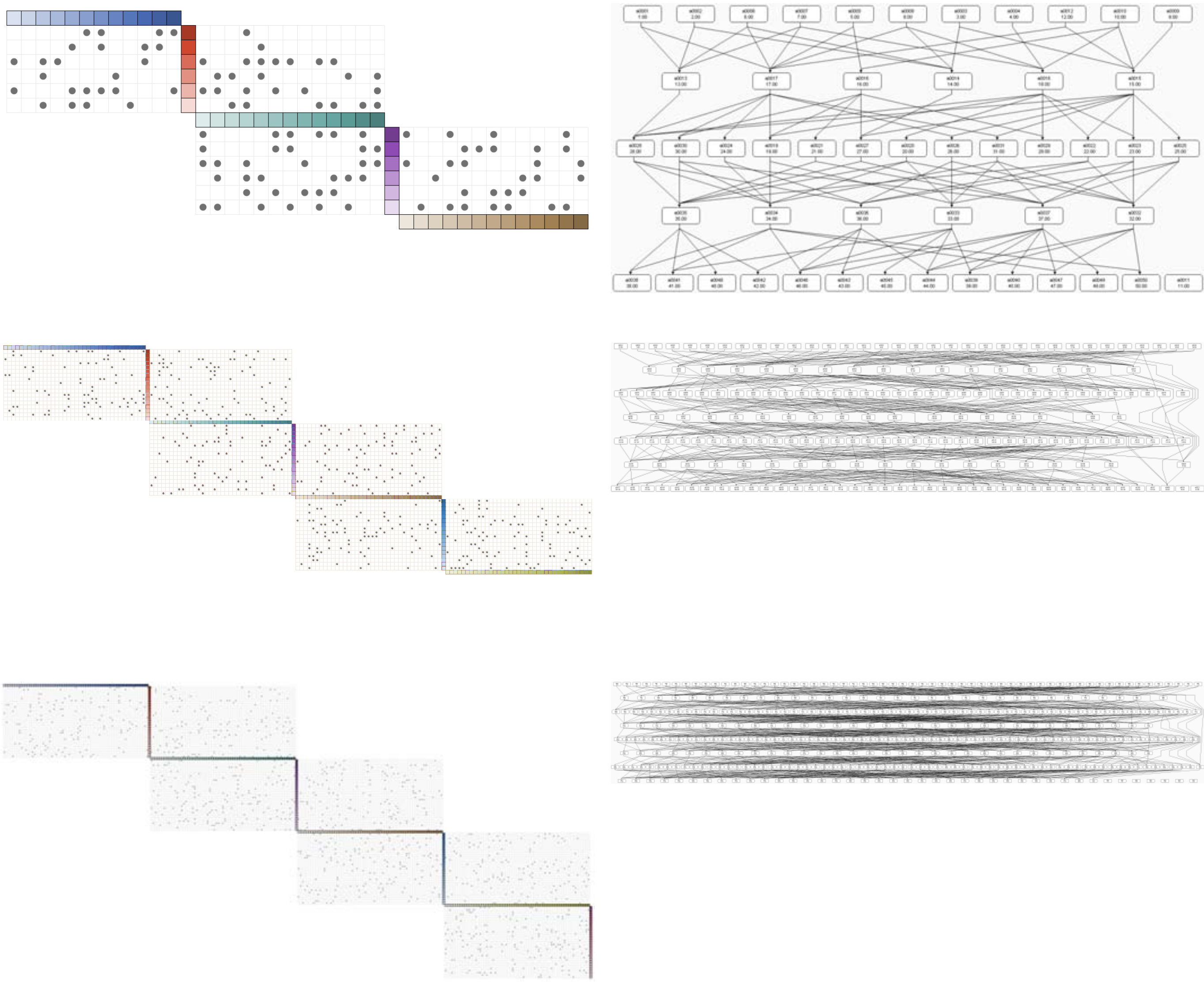

Figure 3: the effect of increasing number of nodes. Quilts on left, node-link depictions on the right. Number of nodes increases from 50 to 200 to 400.

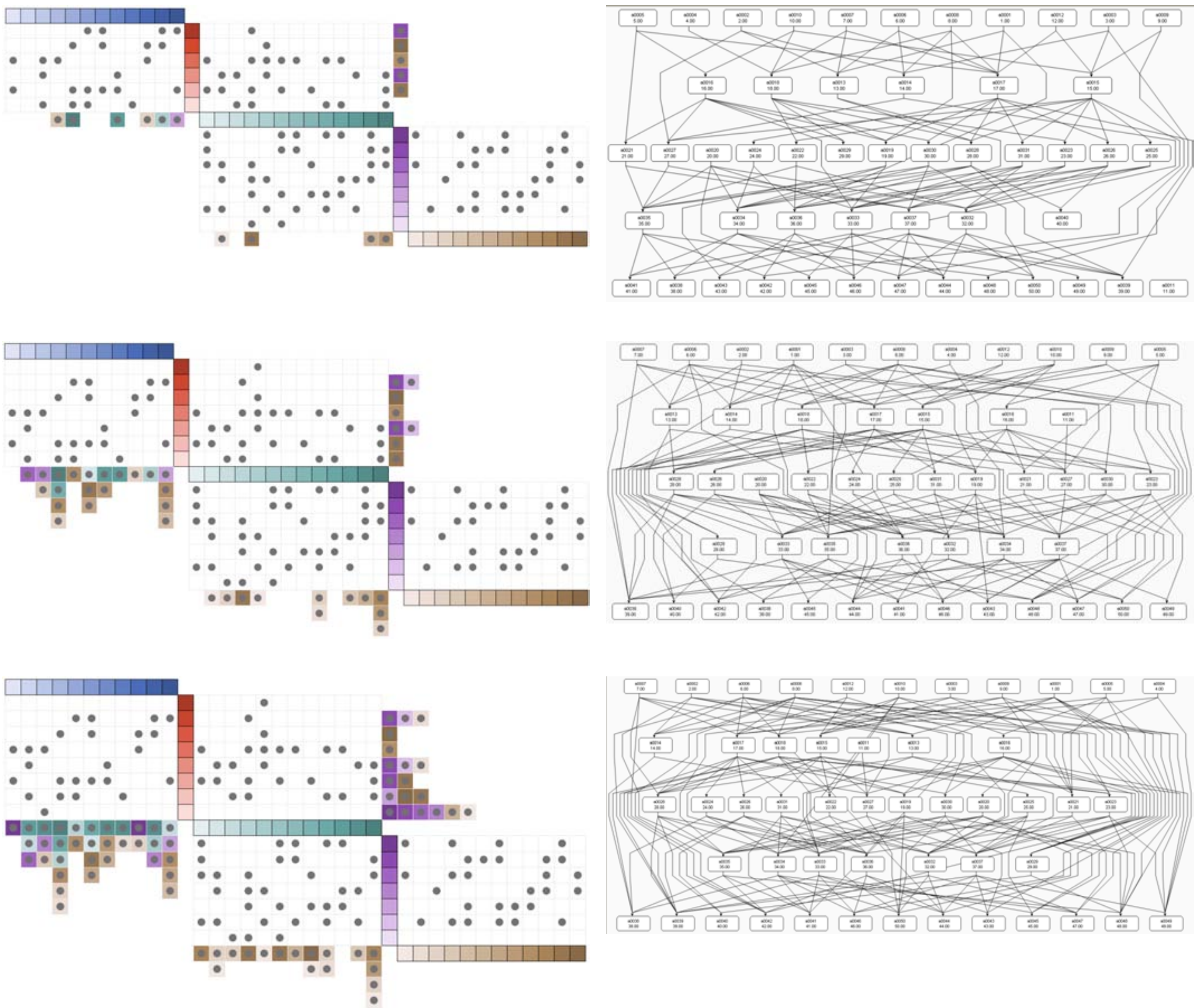

Figure 4: the effect of increasing number of skip links. Quilts on left, node-link depictions on the right. Number of nodes is 50. Skip link likelihood as a percent of number of links increases from 8% to 16% to 32%.

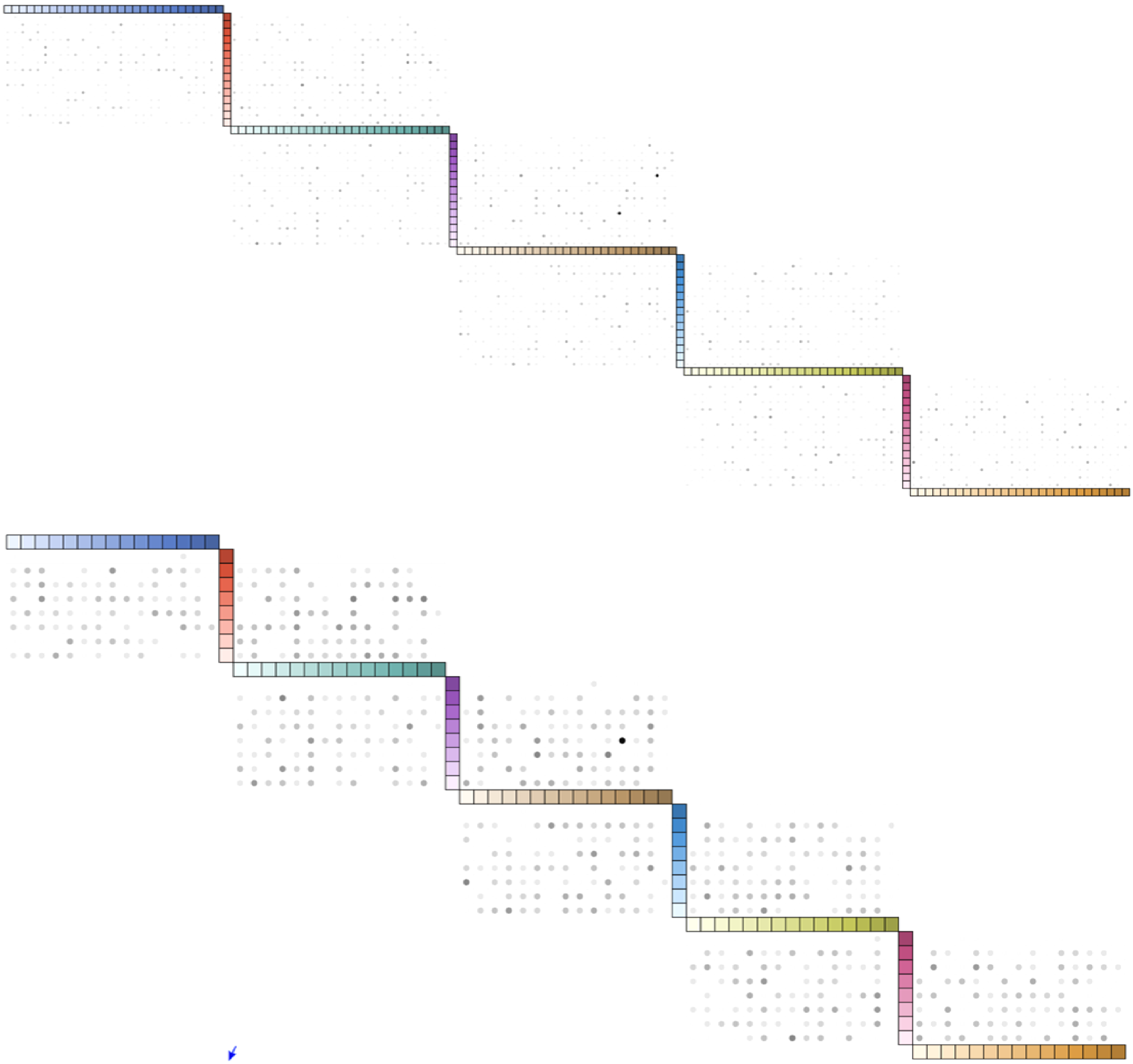

Figure 5: summarization. Here an 800-node graph is summarized with 4 nodes per summarized node (top), and 16 nodes per summarized node (bottom).