

Drawing Graphs Within Graphs: A Contribution to the Graph Drawing Contest 2003

Paul Holleis Thomas Zimmermann Daniel Gmach

Department of Computer Science
University of Passau

holleis@fmi.uni-passau.de zimmerth@fmi.uni-passau.de
gmach@fmi.uni-passau.de

Abstract

This paper is a winning contribution to the Graph Drawing Contest 2003. The task of drawing subgraphs is often underestimated and subgraphs are simply emphasized using different colors or line styles. In this paper, we present an approach for drawing graphs within graphs that layouts the subgraphs first and therefore increases their locality. We introduce connection sets emphasizing relationships between several subgraphs. In a case study, we demonstrate how they can be used to visualize connections between many small network motifs.

1 Introduction

Graphs play an important role in many fields of research and economy, as well as in day-to-day life. Graphs are often huge and complex structures that are difficult to understand. Subgraphs can be a powerful tool for reducing complexity and emphasizing particular aspects. The handling of subgraphs is still not a prominent topic for graph drawing: Usually, subgraphs are not considered for the layout of the entire graph (even if they are known in advance), and only highlighted afterwards. For simple subgraphs this approach suffices but for complex subgraphs their structure is lost.

The topic of the Graph Drawing Contest [4] held at Graph Drawing Conference 2003 was to draw graphs within graphs. The idea is to layout all subgraphs before the entire graph. Thus, subgraphs are easier to spot as their locality is increased. For a single subgraph this approach is straightforward, therefore we concentrate on several subgraphs that are known in advance.

The remainder of this paper is organized as follows. In Section 2 we formalize our approach in general. Section 3 presents a specialized layout algorithm for emphasizing connections between subgraphs. In Section 4, we apply the idea to many similar subgraphs. We close with a discussion of our approach in Section 5.

2 General Approach

First, we present a generic framework for emphasizing multiple subgraphs within one graph, to be able to deal with different tasks and different graphs. Later on, we will specialize this framework and introduce an application.

Create a Summary Graph. In the first step, we create a graph in which each subgraph H_i is represented as a single vertex. We refer to this graph as the *summary graph* $G_S = (V_S, E_S)$. The mapping between the original graph G and G_S is obtained by two functions $f_V : V \rightarrow V_S$ and $f_E : E \rightarrow E_S$. Note, that f_V and f_E may also be used to combine vertices and edges that are not part of a subgraph.

Layout the Subgraphs and the Summary Graph. After the creation of the summary graph, we first layout the subgraphs and then the summary graph. All graphs are layouted independently, so different layout algorithms may be used. If possible, the layout algorithm for the summary graph should use information about the size of the subgraphs.

Apply the Layout to the Original Graph. Next, we combine the layouts of the subgraphs and the summary graph to a layout for the original graph. This is done by replacing vertices that represent a subgraph by the layout of the respective subgraph. The mapping between the original graph and the summary graph can be restored using f_V^{-1} and f_E^{-1} .¹

¹Note that f_V^{-1} and f_E^{-1} are relations and not functions.

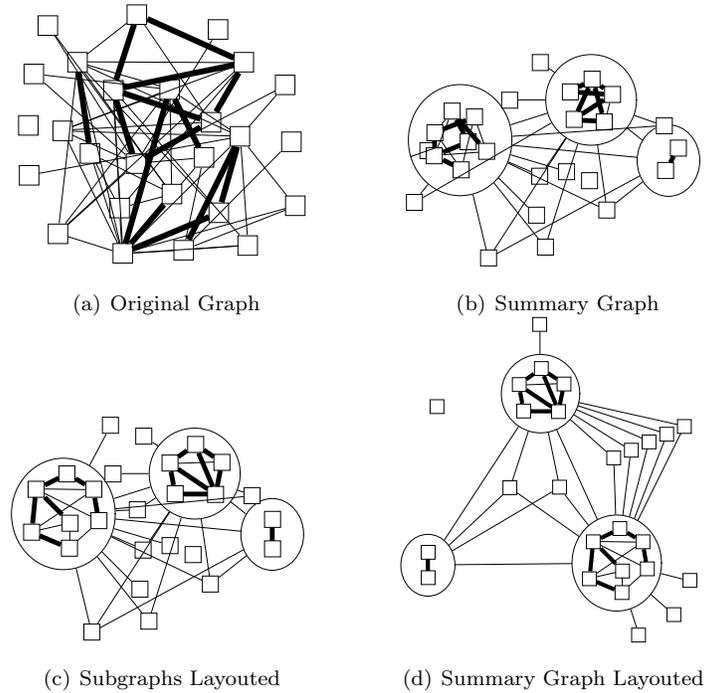


Figure 1: Phases of the General Approach

Optimize the Layout. The resulting layout is not optimal. So, in the final fine-tuning step we reduce the number of crossings and the required space of the layout.

Basically, our summary graphs are special cases of X - Y -graphs [2]. X denotes the structure of an X - Y -graph, and Y the property of some of its nodes. The particular restrictions on Y for a summary graph depend on each individual case.

One graph of the contest represents a social network that evolved from an analysis of organizations involved in drug policy making. The network is an undirected graph where vertices represent organizations and the existence of informal communication chains is mirrored by edges between them. Since the collection of such data always implies uncertainty, *confirmed* relations are given precedence over others.

Figure 1(a) shows this graph, highlighting the *confirmed* edges using bold arcs. Neither the structure of the subgraph nor its relationship with the graph can be seen. The graph is transformed into the summary graph in Figure 1(b). The graph induced by *confirmed* relations consists of three connected components. We decided to take these components as three subgraphs. The subgraphs are layouted individually in Figure 1(c). Finally, the summary graph is layouted and all subgraphs are embedded in its layout in Figure 1(d). Note that the size

of the subgraphs has been taken into account for the layout of the summary graph.

This framework assumes that all subgraphs are disjoint. However, subgraphs that share nodes are not really a problem for our approach. There are three possible solutions:

- Overlapping subgraphs can be *combined* into a new subgraph. This reduces the amount of information present in the choice of subgraphs.
- Vertices contained in several subgraphs can be *duplicated*, creating disjoint subgraphs. Unfortunately edges must be duplicated as well, complicating the layout process.
- Subgraphs can be *partitioned* in overlapping and non-overlapping parts. Each part is then considered a separate subgraph.

3 Connection Sets

Given multiple subgraphs, an interesting aspect is the relationship between them. We present a layout algorithm that emphasizes these relations by identifying *connection sets*. A connection set for some subgraphs contains all vertices that are reachable from all those subgraphs without visiting vertices of any subgraph.

Definition: Let $G = (V, E)$ be an undirected graph, and $H_1 = (V_1, E_1), \dots, H_n = (V_n, E_n)$ be subgraphs of G . The remainder sets are $V_R = V - \bigcup V_i$ and $E_R = E - \bigcup E_i$. A *connection set* is defined as follows:

$$C_{\{i\}} = \{v \mid u \in V_i, \{w_1, \dots, w_k, v\} \subset V_R, \\ \exists \text{ path } u \rightarrow w_1 \rightarrow \dots \rightarrow w_k \rightarrow v\}$$

$$C_I = \bigcap_{i \in I} C_{\{i\}} \quad \square$$

Note, that the definition above allows vertices to be in more than one connection set. For example, if a vertex connects subgraphs H_1, H_2 and H_3 it is contained in seven connection sets $C_{\{1\}}, C_{\{2\}}, C_{\{3\}}, C_{\{1,2\}}, C_{\{1,3\}}, C_{\{2,3\}}$, and $C_{\{1,2,3\}}$. For some graphs it is more intuitive, if such a vertex is only contained in one set. Therefore we define *exclusive connection sets* in which the vertex is only contained in one set $C_{\{1,2,3\}}^e$.

Definition: Let C_I be connection sets of an undirected graph G . We define an *exclusive connection set* as:

$$C_I^e = C_I - \bigcup_{i \in \{1, \dots, n\}, i \notin I} C_{I \cup \{i\}} \quad \square$$

We can now use (exclusive) connection sets to create a layout that emphasizes subgraphs and their relationships to each other. For this we define the

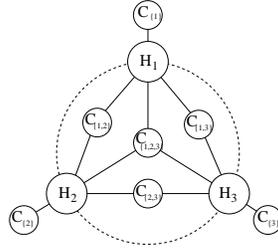


Figure 2: Placement of Connection Sets

summary graph first.

$$f_V(v) = \begin{cases} v_i^H & \text{if } v \in V_i \\ v & \text{else} \end{cases}$$

$$f_E(\{u_1, u_2\}) = \{f_V(u_1), f_V(u_2)\} \text{ for } u_1, u_2 \in V$$

3.1 Layout of the Summary Graph

For the layout of the summary graph we assume that the subgraphs have been layouted in some way and surround those layouts with an ellipse. These are now placed in a circular way, taking their size into account. The oval form of the subgraphs makes it particularly easy to place them on the circle in a pleasant way, i.e. with the same distance to each other. Other geometrical objects might fit better for specific layouts of the subgraphs. The connection sets C_I are put around and between the subgraphs, so neither their vertices nor their adjacent edges interfere with each other. For the layout of the connection sets themselves information about edges that lead from them into the subgraphs is used.

Figure 2 illustrates one possible layout for a graph with three subgraphs H_1 , H_2 and H_3 . For layouts including up to three subgraphs the placement of connections sets does not pose any problems. Although this approach does not seem to scale well with a higher number of subgraphs, it is unlikely that all 2^n connecting sets really show up in the final layout. The reason is that for thin graphs, many connection sets will be empty. For dense graphs, we can concentrate on non-exclusive connection sets connecting zero, one, two or all n subgraphs. This reduces the total number of possible connection sets to $\frac{n^2+n}{2} + 2$.

4 Case Study: Network Motifs

This section applies the general approach presented in Section 2 and the connection sets of Section 3 on a larger graph and a high number of subgraphs.

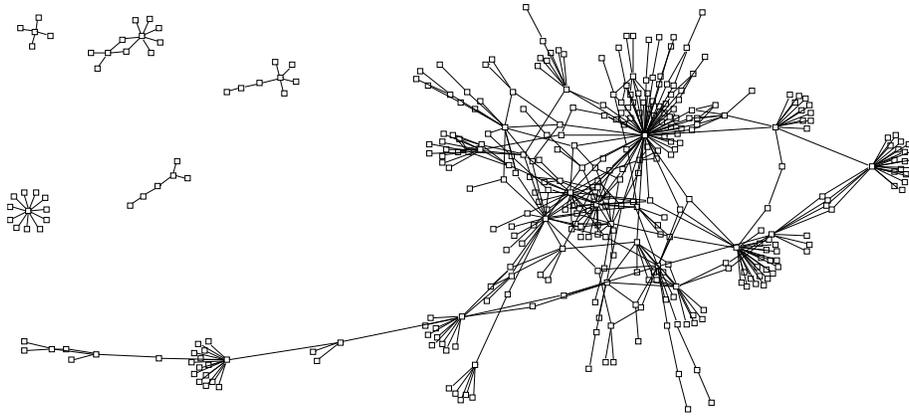


Figure 3: Biological Network layouted with a Spring Embedder [3].

4.1 The Challenge: A Biological Network

We will demonstrate our algorithm on a biological network representing the transcriptional regulation of *Escherichia coli*. The graph consists of 423 labeled vertices and 578 edges. It is shown in Figure 3, layouted using a standard spring embedder algorithm [1]. Basically, the vertices represent operons and transcription factors that interact with each other. These interactions are modeled by edges that carry information on the type of interaction. For a detailed description on this graph and its interpretation we refer to [6].

As subgraphs we will use *motifs*—introduced in [7]. A motif describes small subgraphs that need not necessarily be isomorphic to each other but all match a specific pattern. In biological research, the frequent recurrence of such motifs gives insight into the order of some events and helps to get an overall view of the system, collapsing information that may be spread throughout the whole network into a few nodes.

We will focus on the large connected component of the biological network and leave small components aside. Additionally, we will concentrate on the *single input module* motif (SIM in short) adopted from [6]. The idea behind this motif is that several *operons* are controlled by one single operon, called a *transcription factor*. A search for this pattern in our component of the biological network identified 25 appearances of SIMs. They can be recognized in Figure 3 by their fan-outs.

We show the largest connected component of this graph consisting of 16 appearances of the SIM motif in Figure 4 on the next page. They all have the pattern of one transcription factor controlling several operons in common. Duplicates of vertices appearing in multiple motifs are highlighted. An operon appearing in one motif can, by definition, not be an operon in another motif (as it has exactly one input), but it can play the role of the transcription factor, controlling other operons.

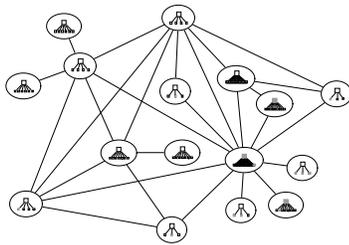


Figure 4: Connected SIMs

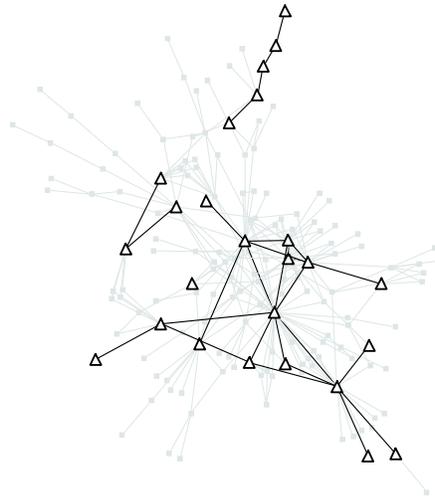


Figure 5: Biological Network with SIMs

4.2 Layout of the Biological Network

We use the 25 identified SIMs as subgraphs for our summary graph. Since the biological network is quite large, we replace the individual subgraphs with a motif-specific geometric shape. This allows us to distinguish between original vertices and vertices representing motifs as well as between several motifs. For the SIM motif, a triangle has been chosen, representing the transcriptional factor on top that controls several operons below. Figure 5 shows a version of the summary graph layouted using the same spring embedder as before.

Applying the connection set method of Section 3 turns out not to be feasible. The number of possible connection sets can be up to 2^{25} . As mentioned before, the actual number of non-empty connection sets is much smaller: In our case there have been 36 non-empty exclusive connections sets. Eleven of those have been connected to exactly one SIM, another eleven to exactly two SIMs and 13 to more than two SIMs. There have been no vertices connected to all SIMs. It is still hardly possible to provide a neat layout for those 25 subgraphs and their connection set. Therefore, we decided to pursue another direction to cope with such a large graph.

Instead of taking each single motif as subgraph, we combine SIMs. As Figure 5 indicates, the SIMs induce a subgraph. This subgraph consists of four connected components. We will visualize the connections between these four components. The justification is that an analyst who is interested in the occurrence of motifs will very probably be interested in the connections between those subgraphs as well.

Figure 6(a) on the following page shows one possible layout resulting from that scenery. This layout also points out one of the strengths of our approach: All those parts of the graph can be layouted using individually chosen algo-

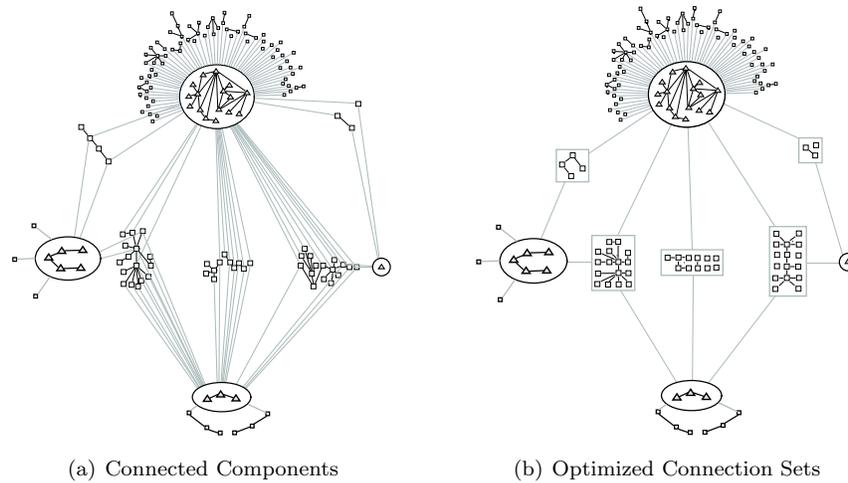


Figure 6: Layouts of the Biological Network using Connection Sets

rithms. In fact, we applied three different methods to create the final layout which we present in Figure 6(a):

- The small components (up to five vertices) have been placed *randomly*, forcing them not to use too much space.
- A *spring embedder* followed by a *crossing-reduction algorithm* has been applied to the subgraph on top and the central connection sets.
- Finally, for the connection set topping the large component, an algorithm has been chosen that works particularly well for thin graphs where most of the vertices are connected to one central vertex. Since there are 73 loosely connected vertices all sharing an edge with the large summary node, this *pincushion* algorithm is ideal.

All those layouts had to take the fixed position of the summary nodes into account.

4.3 Optimization for Large Connecting Sets

For large graphs, with many edges between connection sets and summary nodes, the notion of a summary graph can be extended: The connection sets are considered components and put into (or replaced by) a summary node themselves. Edges incident to a vertex within a connection set are redirected to the corresponding summary node. This considerably reduces the number of edges in the final layout and therefore increases readability. Additionally, the connection sets, having lost many edges, can now be layouted taking up much less space. The optimized layout of our biological network is shown in Figure 6(b).

The central connection sets have been layouted placing nodes on a grid of an appropriate size, reducing arc crossings to a minimum.

5 Conclusion

We presented an approach for emphasizing subgraphs within graphs. Additionally, we described two applications for our approach: One for visualizing connections between subgraphs, and one for highlighting many similar subgraphs. In a final step we combined both techniques. We implemented all ideas presented in this paper and integrated them within a graph editor called GRAVISTO [5]. The GRAVISTO platform has been developed at the University of Passau, Chair for Theoretical Computer Science.

There are situations where our approach produces undesired results, because considering subgraphs for the layout of entire graphs is not always reasonable. As an example take a subway network as the graph, and a connection between two places as the subgraph. Layouting the subgraph first, and the subway network afterwards is not a good idea. First, the vertices of the connection will probably be layouted on a single line or in a circle. And second, the viewer will lose orientation, because the layout does not match his mental map, which is influenced by the geographical distribution of the vertices. The mental map could be restored by enriching the subgraph with additional information about its environment or using graph animation.

Acknowledgments

Prof. Dr. Franz J. Brandenburg provided help and constructive comments on this paper.

References

- [1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, Englewood Cliffs, N.J., 1999.
- [2] F. J. Brandenburg. Graph clustering i: Cycles of cliques. In *Proceedings of the Graph Drawing'97*, volume 1353 of *LNCS*, Berlin, GER, 1997. Springer.
- [3] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.
- [4] Graph Drawing Contest 2003. <http://www.infosun.fmi.uni-passau.de/br/lehrstuhl/GraphDrawing/GD2003/contest.html>.
- [5] Gravisto Project Homepage. <http://www.gravisto.org/>.
- [6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, (31):64–68, 2002.
- [7] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, (298):824–827, 2002.