Effectively Visualizing Large Networks Through Sampling

Davood Rafiei
Department of Computing Science
University of Alberta
drafiei@cs.ualberta.ca

Stephen Curial
Department of Computing Science
University of Alberta
curial@cs.ualberta.ca

ABSTRACT

We study the problem of visualizing large networks and develop techniques for effectively abstracting a network and reducing the size to a level that can be clearly viewed. Our size reduction techniques are based on sampling, where only a sample instead of the full network is visualized. We propose a randomized notion of "focus" that specifies a part of the network and the degree to which it needs to be magnified. Visualizing a sample allows our method overcome the scalability issues inherent in traditional visualization methods. We report some characteristics that frequently occur in large networks and the conditions under which they are preserved when sampling from a network. This can be useful in selecting a proper sampling scheme that yields a sample with similar characteristics as the original network. Our method is built on top of a relational database, thus it can be easily and efficiently implemented using any off-the-shelf database software. As a proof of concept, we implement our methods within a system called *ALVIN* and report some of our experiments over the movie database and the connectivity graph of the Web with 178 million nodes and over 800 million edges.

Categories and Subject Descriptors

I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

visualizing the Web, large network visualization, network sampling, searching a network

1. INTRODUCTION

The extensive growth of the Internet within the past few years has led to a proliferation of very large networks; examples include bibliographic collections, biological networks, market basket data, the Internet (both in the router and the inter-domain layers), and the World Wide Web. Although the collection and the storage of such data has become relatively straightforward, effectively analyzing data has proven to be more difficult. Visual display of networks, in particular, can lead to both better understanding and clear presentation of patterns that can often be hidden [20]. Alfred Crosby, the historian, lists "visualization" as one of the two processes that has led to the explosive growth of modern science; the other process is "measurement" [9]. Visualizing "large" networks, however, can be quite challenging if not impossible. This is due to the limitations of the screen, the complexity of layout algorithms and the limitations of human visual perception. A good layout algorithm (eg. Spring layout) can easily take quadratic time assuming that the network fits in memory. The graph structure of the Web, for instance, is far too large to hold in the memory of most desktops let alone visualize it.

To gain insight into the complexity of the problem, consider the graph structure of the Web at the domain level as shown in Figure 1-a. This network is relatively small, having only 224 nodes, but it is still not easy to find any interesting patterns. Colouring the largest *Strongly Connected Component* [1](SCC), as is shown in Figure 1-b, singles out some of the domains that are not in the SCC. One such domain, for instance, is *Vatican City* (va), linked by a large number of domains in the SCC but is not linking back to any domain in the SCC. Even in this graph, it is not easy to see the connectivity structure of many of the domains in the SCC. Scaling up the visualization to a graph of the Web with millions of nodes at the site level or hundreds of millions of nodes at the page level is quite challenging if not impossible.

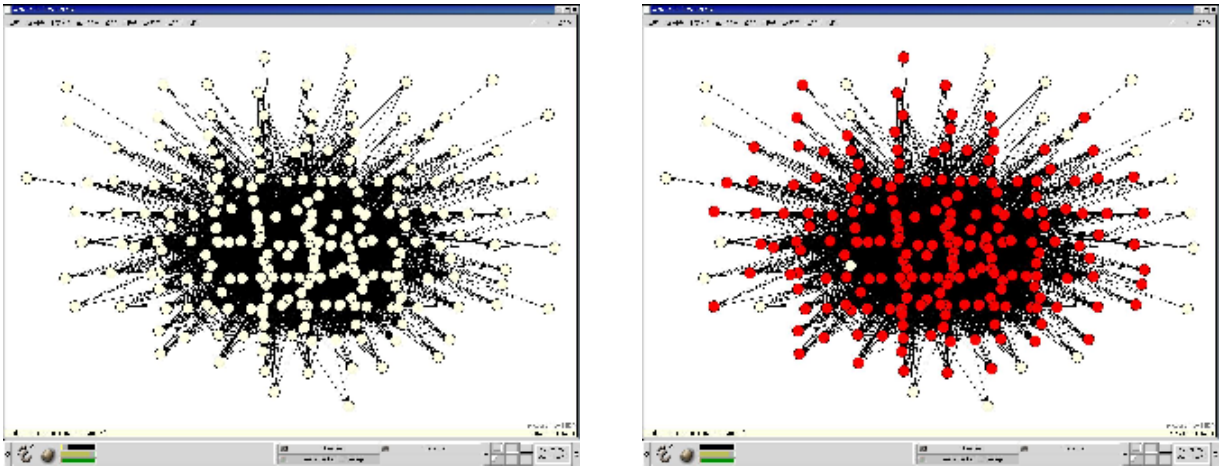
Our proposed alternative in this paper is to refrain from visualizing the entire network. At the core of our methods is sampling. We sample the network and only visualize the sample. Even though the network can be quite large, the size of the sample can be adjusted to match the limitations of the visualization environment. We study some of the topological properties of a network that are preserved in a sample and show that a relatively small sample, if collected carefully, can still show some of the patterns that are inherent in the entire network.

As our second contribution, we develop a notion of "focus", one can set, to bring into focus only part of the network that needs to be explored in greater detail. This is done in the context of the full network. If no focal point is set, the network is sampled uniformly. In the presence of a focal point, the sampling is biased toward that focal point, thus the visualization emphasizes the focal point and its neighbourhood in the network.

In this paper, we propose several sampling-based schemes for both focusing the search and visualizing networks which are too big to be fully visualized. We formalize a notion of

[1]A *strongly connected component* of a graph is a set of nodes such that for any pair of nodes u and v in the set, there is a path from u to v.

WWW2005, May 10–14, 2005, Chiba, Japan.
.



(a) (b)

Figure 1: *(a) The connectivity network of the World Wide Web at the domain level, and (b) the same network with the SCC coloured red.*

focus for both networks with directed and undirected graph structures. We further extend this formulation to the case where edges in the underlying graph structure are weighted. Anecdotal evidence is provided to show that these schemes can be quite useful.

We have built a prototype, named *ALVIN* [2], that implements the ideas described in this paper. As a proof of concept, we run ALVIN over the movie database and the connectivity graph of the Web. We abstract the Web graph into three layers: *domain*, *site* and *page*, and demonstrate some of our experiments over these layers.

The rest of the paper is organized as follows. After motivating our work in Section 2, we discuss issues related to sampling a network in Section 3. Our proposed scheme for visualizing and expanding a network is discussed in Section 4, and our notion of focus is presented in Section 5. Section 6 presents some implementation details and our experimental results. Section 7 reviews the related work, and Section 8 concludes the paper.

2. A MOTIVATING EXAMPLE

Consider the connectivity network of the Web with each node describing a Web page and each edge describing a hyperlink. Due to the huge size of the network, there is no hope visualizing the entire network. However, the problem becomes simpler if we can turn our focus on a few specific Web pages. Suppose we are interested in all Web pages in a particular site such as the *CS Department* home page at the *University of Alberta* [3] (CS@UofA). There can be already too many pages after focusing on a single site. Figure 9 shows 800 edges selected randomly from the Web graph with the condition that one endpoint of each edge is a page from our desired site. For clarity, we remove singleton edges that are not connected to any other components. The resulting network emphasizes the connections between our desired pages and the rest of the network, highlighting some of the local pages with large interconnections to the rest of the Web. In particular, it shows some of the pages at CS@UofA such as the department home page, the home page of the Graduate Student Association and the system support pages [4] all with strong ties to the rest of the pages in the CS@UofA site. It also shows other Web pages such as the university home page with large link connections to many of the pages in CS@UofA.

Suppose we also want to visualize the Web graph as a whole without emphasizing a specific site, perhaps identifying some of the general patterns. To reduce the size of the network, we may want to visualize the Web graph at the site level, with each node describing a site and each edge denoting a link from a page in one site to a page in another site. Figure 10 shows the result of randomly selecting 2000 edges from the set of edges between sites and removing unconnected singleton edges. The graph clearly shows some of the *authorities* such as *Netscape*, *Microsoft*, *Adobe* and *Yahoo* and some of the sites with large link collections, referred to as *hubs*, such as *Yahoo Directory* and *Fisher-Rosemount Companies* (frco). Some sites such as *AOL members* and *geocities* enjoy a large number of both incoming and outgoing links. There is also a dense irregular connection between sites that do a link exchange such as *Infospace* and *Link Exchange*. A strongly connected component (coloured red in the picture) is formed between some local sites of *Infospace*.

3. SIMPLE RANDOM SAMPLING OF A NETWORK

In this section, we discuss several ways of sampling a network and some of the characteristics of the original network that can be observed in the sample. In the next section, we formalize these sampling schemes in the form of some growth processes and develop a general model for visualizing a network.

Given a network $G(V_G, E_G)$, any subgraph of G can be treated as a sample of the network. Clearly, there are different ways of taking a subgraph and as a result there are

[2] The name *ALVIN* stands for *Alberta system for Visualizing Large Networks.*

[3] *www.cs.ualberta.ca*

[4] www.cs.ualberta.ca/operations

many different sampling strategies. We use the following three methods for obtaining a simple random sample of a network. Independent of the strategy used for sampling, we let $S(V_S, E_S)$ denote a simple random sample of G.

SRS_1: Take a simple random sample of the nodes, V_S, and let $S(V_S, E_S)$ be a subgraph of G induced by V_S.

SRS_2: Take a simple random sample of the edges, E_S, and let $V_S \subset V_G$ be the set of nodes incident to at least one edge in E_S.

SRS_3: Take a simple random sample $S'(V'_S, E'_S)$ using SRS_2 and let S be a subgraph of G induced by V'_S.

There is a caveat when sampling from nodes; unless the network is very "well-connected," the resulting sample would be quite sparse. This is not hard to verify; given a network with N nodes and k edges per node on average, if we pick only n nodes randomly, each node in the sample will be connected to only $k\frac{n}{N}$ other nodes on average. This number is expected to be almost zero unless the sample includes a large fraction of the nodes, k is large or both.

Sampling from edges instead may be more desirable because the sampled network is no longer sparse. This sampling is unbiased toward edges but not toward nodes. Nodes with large in- or out-degrees are more likely to be in the sample, and paths of length greater than one are likely to form between them. This is not as problematic as it may look since those nodes are likely to form the backbone of the network and it is good to have them in the sample.

There are other strategies for sampling a network. Some of those can be found elsewhere [21].

3.1 Using Sampling to Visualize Network Topology

There are a number of traits which are found in every network, and can be useful in describing the general topology of a network. These include the degree distribution, connected component size distribution, characteristic path length, clustering coefficient, etc. Some of these traits can be preserved when sampling from a network. We study the degree distribution and the connected component size distribution, two of the properties that appear to be important in visualizing network topologies. The degree distribution of the Web graph, for instance, provides stratified counts of the degrees, differentiating hubs and authorities from other pages [7]. This property in turn can be useful in a visualization, as evidenced in our motivating example. The component size distribution is another important visual feature that can be representative of a network (e.g. see the results reported for the Web graph [10]), and we often want it to be preserved in a sample. We discuss these two features in the context of the movie database from *IMDb* [5], where each actor is represented by a vertex and there is an undirected edge between two actors if the actors are cast together in the same movie.

For average path length and clustering coefficient (definitions can be found in Watts [31]), it is not clear if these features can be preserved in a sample. Consider G_1 as a complete graph and G_2 as a complete bipartite graph. The clustering coefficients of G_1 is 1 and G_2 is 0. A small sample of both graphs taken using SRS_2 can give a clustering coefficient of 1 for both graphs if, for instance, the selected edges are not connected. Increasing the sample size is expected to decrease the clustering coefficient of G_2 but this decrease is not monotonic. However, increasing the sample size is expected to decrease and then increase the clustering coefficient of G_1. Therefore, for relatively small to medium-sized samples, the clustering coefficient is more dependent on the sampling strategy and size than the structure of the original network. Similarly, the average path length in a sample is also non-deterministic and can largely vary from one sample to the next. Next, we report results obtained by sampling the movie database and analyzing the samples.

3.1.1 Degree Distribution

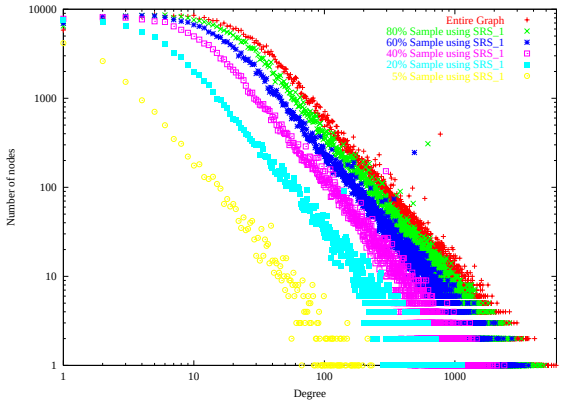


Figure 2: *Degree distribution using SRS_1 for sampling.*

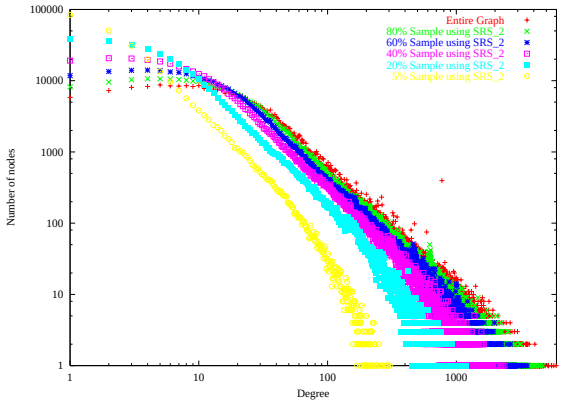


Figure 3: *Degree distribution using SRS_2 for sampling.*

Figure 2 shows the degree distributions of the movie database with the sampling strategy fixed to SRS_1 and the sample size varied from 5% to 100%. The degree distribution remains relatively close to the entire network, even for a small sample. The same trend can be observed when we change the sampling strategy to SRS_2, as shown in Figure 3.

3.1.2 Component Size Distribution

Figures 4 and 5 show the component size distributions of the movie database, taken using strategies SRS_1 and SRS_2

[5] *IMDb* - Internet Movie Database (www.imdb.com)

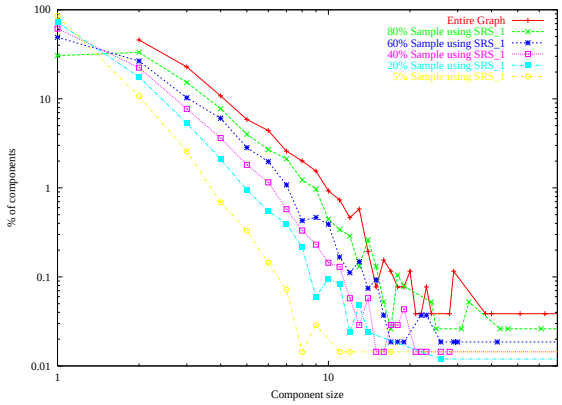


Figure 4: *Connected component distribution using SRS_1 for sampling.*

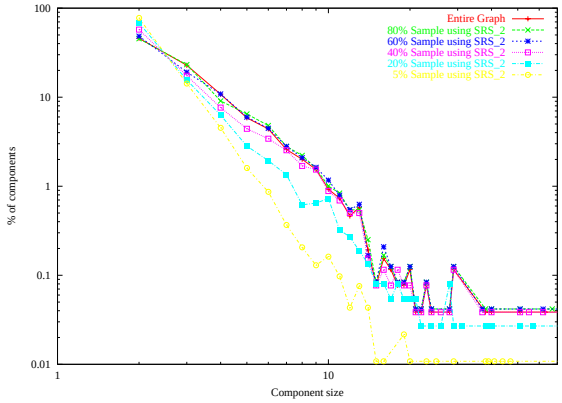


Figure 5: *Connected component distribution using SRS_2 for sampling.*

respectively. The sample size in both graphs is varied from 5% to 100%. The component size distribution remains relatively close to the entire network, regardless of the sampling method. SRS_2 seems to preserve the component size distribution, closely resembling the original data.

Finding a relationship between the distribution of component sizes in a sample and the number of components in the entire network is not new. For transitive graphs [6], in particular, Frank has shown that if we sample the network using SRS_1, the resulting network can be used to find an unbiased estimate of the number of connected components of the entire network [12].

Theorem 1. *Let the parent graph be transitive, and suppose $S(V_S, E_S)$, a simple random sample taken using SRS_1. Let $v = |V_S|$. If $K_r(S)$ denotes the number of connected components of size r in the sample, then an unbiased estimate of the number of connected components in the parent graph is given by*

$$\sum_{r=1}^{M} (1 - C_r) K_r(S)$$

where

$$C_r = (-1)^r \binom{N - v + r - 1}{r} \binom{v}{r}^{-1},$$

N is the number of nodes in the parent graph, $M \leq v$ is a constant and the parent graph has no connected component of size larger than M.

Both the proof and the variance of this estimate is given by Frank [12]. Abusing the theorem, we tried using SRS_2 with Frank's estimate on synthetic data. Our synthetic data included graphs consisting of both complete connected and complete bipartite components. The component sizes were generated randomly and varied from 4 to 80. The results showed that Frank's estimate used with SRS_2, sampling only 25% of the edges, could accurately estimate the number of components with an average error of less than 8%.

4. NETWORK GROWTH

Despite the encouraging results of our sampling methods, the original network can be large, and visualizing a small sample that can preserve some of the desired topological properties of the network may not be feasible. To address this problem and to provide a navigation scheme, we develop several growth processes, collectively referred to as *network growth*, that allows one to interactively visualize a network.

In an interactive fashion to some degree similar to Web browsers, our visualization starts with a small subset of the network which may include a set of hand-picked nodes and edges or the result of a query. The visualization may proceed towards the goal by iteratively growing the initial set. This is useful for narrowing down the visualization to some of the interesting elements when the network is too large to be fully visualized. A novelty of our method is the way the network, currently displayed on canvas, is expanded. Our method uses user-controllable parameters to describe how and to what degree the network must be expanded. The expanded network often has more detail about the elements

[6] A graph is *transitive* if there is an edge between every connected pair of vertices.

being studied yet is small enough to be visualized and internalized. After a few layers of extension, the network may become too large; this may be an indication that the browsing should switch to another small subset before it can continue.

Let $G(V_G, E_G)$ be the network that needs to be visualized and $C(V_C, E_C)$, a subgraph of G, be the network that is currently displayed on canvas. Our model iteratively picks nodes from $V_G - V_C$ and edges from $E_G - E_C$ and adds them to C, thus expanding the network on canvas with respect to G. We discuss several ways of expanding a sample of a network, formalize these sampling schemes in the form of some growth processes and develop a general model for visualizing a network.

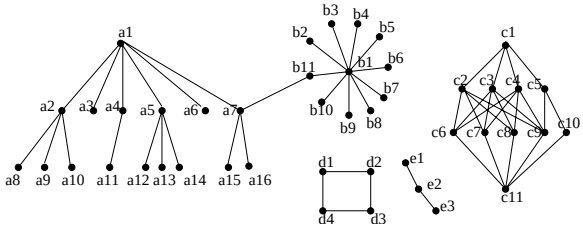


Figure 6: *A network instance.*

4.1 Global Growth

Sometimes we want to gain some insight into the general connectivity structure of the network without specifying a pivotal point; or we might be interested in only part of the network but want to browse this part in the context of the entire network. We may achieve this by taking a simple random sample of the network and visualize the sample. One such sample can provide the general connectivity structure of the network and maybe some common patterns without emphasizing one specific part. Clearly, the larger the sample, the more accurate the estimates and also the more detailed the visualized network; though a detailed sample may not always be clearly visualized.

Definition 1. Let C be a subgraph of a parent network G. A *global growth* of C with respect to G adds to C a simple random sample of G taken using one of the sampling strategies from Section 3.

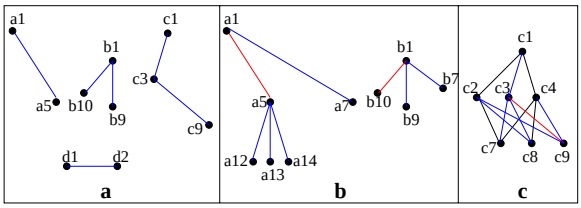


Figure 7: **(a)** *a global growth,* **(b)** *a local growth with initial edges (a1,a5), (b1,b10) and* **(c)** *a local growth with initial edge (c3,c9).*

Example 4.1. Figure 6 shows an instance of a network with 45 nodes and 55 edges. A simple random sample obtained using SRS_2, with only six edges picked from the random ordering shown in the appendixis displayed in Figure 7-a. This sample, consisting of 11% of the edges, shows some of the components of the parent network; it has the same number of connected components as the parent network even though the components are not necessarily the same.

4.2 Local Growth

We often know some of the nodes and maybe some of the edges of a network and wish to find more related nodes and edges somehow related to our starting set, or we may like to find out how our starting set fits within the building blocks of the entire network. This can be done through sampling from the network surrounding C and adding the sample to the canvas. The sample includes some of the edges that glue C to the rest of the network G.

Definition 2. Let C be a subgraph of a parent network G, and let $I(V_I, E_I)$ be the subgraph of G such that E_I is the set of edges with one endpoint in V_C and the other endpoint in $V_G - V_C$ and V_I is the set of nodes incident to any edge in E_I. A *local growth* of C with respect to G adds to C a simple random sample of I taken using one of the strategies from Section 3.

Our local growth generalizes a sampling method, often referred to as *snowball sampling*, which is typical of a link-tracing design where a simple random sample or stratified random sample of units is selected and all other units linked to the initial sample are included or observed [30]. The initial set in a local growth is not necessarily picked randomly; instead, it can be the result of a user query. Furthermore, a local growth does not necessarily include all edges linked to the initial set since this can be too large.

Example 4.2. Figure 7-b shows the result of a local growth after hand-picking the edges $(a1, a5)$ and $(b1, b10)$ from the network in Fig 6 and adding 6 more edges selected using SRS_2 through a local growth. For our edge selection, we again use the random ordering in the appendixbut only add edges with one endpoint in {a1,a5,b1,b10}. As another example, Figure 7-c shows the result after hand-picking $(c3, c9)$, doing a local growth using SRS_3 which adds 6 more edges (these edges are coloured blue) and further extending the graph to include edges with both endpoints already selected (these edges are coloured black). Compared to a global growth that shows more of the structure of the entire network with less resolution, a local growth depicts a specific part of the network in greater detail but with less information about the network as a whole.

4.3 Mixed Growth

A local growth can be combined with a global growth at a user-specified rate to provide a more balanced mixture of the two. Under this scheme, called a *mixed growth*, the network is sampled as follows: with some probability we perform a local growth and with the remaining probability we perform a global growth. A mixed growth provides a spectrum of sampling schemes with local and global growths as the two ends of the spectrum.

4.4 Wiring

Sometimes we have our desired nodes on canvas but wish to visualize the interconnections between them in greater detail. A solution is to add more edges between the nodes

on canvas. We call this process *wiring*. In the extreme case, a wiring can add all the edges between nodes on canvas. However, this may clutter the visualization, obscuring the details. Therefore, a user-specified parameter may control the degree of wiring.

4.5 Rewiring

Since selecting edges is a random event, there are many possible wirings, and we may wish to view more than one possible wiring of the nodes on canvas. Through the process of *rewiring*, all edges on canvas can be removed and the nodes on canvas can be wired again. This may reveal properties that may not have been displayed by the original wiring.

5. FOCUSED BROWSING: A GENERAL MODEL

The local growth provides a method to focus on a specific part of the network, but the part of the network we want to focus on may not fit on canvas. Furthermore, we may not want to display the area we wish to focus on and rather use it to direct the growth. We introduce a more general notion of *focus*, independent from the network on canvas, that can be used to narrow the visualization to a desired part of the network, reducing both the size and the complexity of the visualized network. Our notion of "focus", referred to here as *focal point*, formulates to some extent our *interest* at browsing. For instance, if we are only interested in a few nodes, then these nodes can form our focal point; or the focal point may be set to the network currently on canvas or only part of it where further details are needed. Without loss of generality, our browsing goal is to visualize the focal point in the context of the entire network.

The following scenario shows how this model can be useful. Consider the connectivity graph of the Web where nodes represent Web pages and edges describe the hyperlinks between pages. Suppose we are interested in the connectivity of pages on a specific topic say *surfing*. We can set the focal point to include all pages that mention the term 'surfing' in their contents. There can be many more pages on this topic than what we can fit on canvas, thus we may visualize only a subset of these pages. If we expand the visualized set by adding pages that either link to a page in the initial set or are linked by a page in the initial set, the resulting set is shown to include the most prominent sources of primary content known as *authorities* and high-quality guides and resource lists known as *hubs* on the search topic[19].

It is not hard to integrate this notion of "focus" into our visualization scheme. Since our visualization is based on sampling, the network is sampled and only the sample is visualized. In the presence of a focal point, the sampling is biased toward this focal point.

5.1 Formal Model

Given a network $G(V, E)$, a *focal point* is formally a subgraph $F(V_f, E_f)$ where $V_f \subseteq V$ and $E_f \subseteq E$. In the absence of a focal point, F is naturally G, meaning that we are interested in the entire network.

A transition from one step of the browsing to the next step is described using a growth process. A *growth* describes how the network on canvas must be expanded using a sample of the parent network and with respect to a focal point F. A *growth* more formally is a mapping from the set of subgraphs of G to the set of subgraphs of G. The mapping takes two real number parameters that control the degree of bias towards the focal point.

Definition 3. Let C and F be subgraphs of a parent network G, and let $I(V_I, E_I)$ be the subgraph of G such that E_I is the set of edges with one endpoint in V_F and the other endpoint in $V_G - V_F$ and V_I is the set of nodes incident to any edge in E_I. A *focused growth* at rate (r, s), where $r, s \in [0, 1]$, of C with respect to the focal point F and the parent graph G adds to C simple random samples of I, G and F with sample sizes respectively proportional to $s(1-r)+r(1-s)$, $(1-r)(1-s)$ and rs, each sample taken using one of the strategies from Section 3.

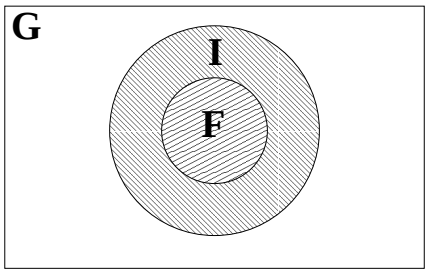


Figure 8: *Focus set in the context of network G.*

Figure 8 shows a graphical picture of the sets F, G and I. A focused growth combines two *simple random samples* with a *snowball sample* at a user-specified rate. An interpretation of the parameters r and s is that if r denotes the probability of picking an endpoint from V_F, then $1-r$ is the probability of picking the same endpoint from V_G. Similarly, if s denotes the probability of picking the other endpoint from V_F, then $(1-s)$ is the probability of picking it from V_G. If we set the focal point to the network on canvas and $r = 1$ and $s = 0$, a focused growth simulates the local growth of Section 4.2. A focused growth also simulates the global growth of Section 4.1 if we set the focal point again to the network on canvas and $r = 0$ and $s = 0$. Varying the values of the parameters r and s, we can obtain other variations of a network growth.

5.2 Directed and Weighted Networks

It is not hard to extend our proposed schemes to both directed and weighted networks. For a directed network, we may fix in advance the fractions at which a source and a destination must be selected from V_F. One simple setting, for instance, is to set the ratios to 50/50 or some other constant. An alternative is to allow the ratios to be set at the time of the browsing using additional parameters.

In a weighted network, often the weight of an edge describes the strength of the relationship between the two endpoints. In a commuting network, for instance, each edge may be weighted to indicate the frequency of travels made in a day. If the network consists of more than one level of abstraction, each node or each edge in a more general layer may be weighted and the weight may aggregate multiple nodes or edges from a more specific layer. For instance, the connectivity graphs of the Web on the domain and site levels can be seen as aggregations of the Web graph on the page level. If the weight of a node or an edge is treated as an

indication of its importance, we want to bias the visualization towards highly-weighted edges. This is again possible within our sampling framework by replacing a simple random sample with a weighted sample.

6. EXPERIMENTS

ALVIN, our current prototype implementing these ideas, has the following highlights:

- It uses the DB2 relational database as its back-end data storage and querying engine. It makes no assumption on the size of the network and the back-end relational database can efficiently handle very large data sets.
- It provides an interface for both focusing and expanding the network on canvas. It allows the user to interactively expand the graph on canvas using parameters r, s and *the size of the sample*. Requests that arise from user interactions are mapped to SQL statements and are directed to the back-end SQL engine for an efficient evaluation.
- It is developed in C++ using the LEDA class library [22] and makes use of the layout and graph algorithms that are available in this library.
- Network abstraction and hierarchical views are supported by creating tables and views in the relational database.

We ran ALVIN over two data sets: (1) the movie database from *IMDb* and (2) the linkage structure of a snapshot of the Web from *Internet Archive* [7]. In the movie network, each actor was represented by a vertex and there was an undirected edge between two actors if they were cast together in the same movie. In the Web connectivity data set, each vertex denoted a Web page and each directed edge denoted a hyperlink. Both networks were stored as relational tables. For the Web graph, we also constructed two hierarchical views of the data in the site and the domain levels. These graphs were weighted with the weight of an edge representing the number of links from one site (domain) to another. For efficiency reasons, these views were pre-computed and physically stored. Next, we report some of our results with these two data sets.

6.1 Web Graph

As our first experiment, we placed all sites in the `.org` domain in our focus set, implemented as a relational table, and did a focused growth of the network at rate (1,0), selecting 3000 edges. Figure 11 shows the result after removing all connected components consisting of four or less nodes. Some sites from the `.org` domain such as *w3.org*, *pbs.org*, *eff.org* and *unicef.org* can be easily identified because of their dense connections with the rest of the network. The figure also shows sites from `.com` domain such as *members.aol.com*, *geocities.com* and *adobe.com* that have dense connections with sites in the `.org` domain.

As our next experiment, we used the same focus set but this time did a focused growth at rate (1,1), selecting 1000 edges. Figure 12 shows the result again after removing (unconnected) singleton edges. Despite using the same focus set, we obtained a different set of nodes with another interesting pattern between them. The result included sites such as *AMC Cancer Research Center* [8], *American Academy of Allergy Asthma & Immunology* [9] and *American Academy of Pediatrics* [10], all in the `.org` domain, with a relatively dense connections between them.

6.2 Movie Database

Experiments were also conducted using the movie database. One interesting experiment was to add a number of "famous" actors to the canvas and explore their relationships with the rest of the actors. We hand-picked 38 actors and added to the focus set along with all the inter-connecting edges between them. We did a focused growth at rate (1,1), randomly selecting 400 edges, and added them to the canvas. A few of the actors from the focus set, highlighted in the green rectangles, are shown in Figure 13. We then did a focused growth at rate (1,0), randomly selecting 500 edges, and added them to the canvas. Some "famous" actors such as George Clooney and David Arquette who were not in the focus set were identified by this growth process. These actors are shown in Figure 13 with blue ovals.

7. RELATED WORK

It has been noted that *layout*, *abstraction*, *focus* and *interaction* form the basis of visualizing large networks [23]. Our work addresses the issues of focus, interaction and partly abstraction; for the layout, we use standard *force-directed* layout algorithms [11].

There has been past work on layout and encoding schemes that can scale-up to large trees or more specific graphs. In particular, Munzner [24] constructs spanning trees to represent the structure of a class of graphs with more tree-like structures, referred to as quasi-hierarchical graphs. The resulting tree is drawn inside a ball with fisheye distortion used to provide a focus-context view. Abello et al. [1] propose a hierarchical partitioning of the nodes based on characteristics such as the geographical locations that the nodes may represent. Using these partitions, different navigation and visualization schemes can be constructed [3]. Our work is different from these in that we don't make any assumption on the structure of the network or the characteristics of the nodes. When such information is present, it is easy to integrate other abstraction techniques (e.g. graph slices [2]) within our framework, using relational tables and views.

The work on general multiscale abstraction methods allows one to visualize either the global structure or the smaller components of a large network (e.g. [5, 18]). These methods usually do a clustering of the network and provide a coarser visualization between the clusters and a finer visualization within each cluster but not both at the same time. Gansner et al. [14] propose a notion of a hybrid graph which allows the region of interest to be viewed in a finer level and within the coarser graph. Other abstraction techniques include, but are not limited to, the work of Noik [25], Plaisant et al. [29] and Herman et al. [17]. Our work is orthogonal to all these abstraction methods; our methods are applicable to coarser

[7] The *Internet Archive* is a public nonprofit organization that offers access to historical collections that exist in digital format, including the entire Web. (www.archive.org)

[8] amc.org

[9] aaaai.org

[10] aap.org

views of a network when the coarser view is still too large to be fully visualized. Our use of sampling for distortion makes our work different from standard fisheye distortion techniques [13]. Our design decisions allow an easy integration of our method with other abstraction techniques and focusing methods.

Related to sampling from a large database, a number of algorithms have been proposed for efficiently sampling from a single table and also from the results of set union, intersection and join [26, 8]. A survey of these techniques before 1994 is given by Olken [27]. Sampling is now supported in major commercial databases and is also part of the recent SQL standard [15].

Related to our work is also the more general work on analyzing social networks (e.g. [31], [4]), mining graphs [28], URL sampling [16, 6] and analyzing the graph structure of the Web [7].

8. CONCLUSIONS

A new probabilistic approach for effectively searching and visualizing large networks is proposed, where only a sample instead of the entire network is visualized. There is no concept of a unique visualization of a network in this scheme; instead there are many possible visualizations, each corresponding to some random sample of the network. The effectiveness of a sample and, as a result, a visualization that is based on that sample depends on the presence of some of the desirable patterns of the parent network in the sample. We have provided some evidence to show that indeed such patterns are preserved in a sample. Given the limitations of the screen and the size of a sample, our proposed scheme allows the search to be localized, thus increasing the ratio of sample size to the size of the desired network and removing possible biases due to the sample size.

Our work touches some of the problems related to visualizing a sample of a network. There are a number of issues that are open to further research:

- Even though sampling has been largely used to approximately answer aggregation queries on large data sets, there is not much work on finding sampling strategies that can preserve either the local or global properties of a network. Further studies on the subject can lead to more effective visualization schemes.

- Our work treats visualization as an incremental process that may lead to the goal after a number of growths. After each growth, a layout algorithm must be invoked to properly place the network on the canvas. A new layout may not be coherent with the old one and the elements in both layouts can be placed in different locations of the screen. Further research may look into algorithms that can preserve the locality of the nodes and still generate an effective layout after each growth.

9. REFERENCES

[1] J. Abello, I. Finocchi, and J. Korn. Graph sketches. In *Proc. of the IEEE Symposium on Information Visualization*, pages 67–70, San Diego, October 2001.

[2] J. Abello and J. Korn. Visualizing massive multi-digraphs. In *Proc. of IEEE Symposium on Information Visualization*, pages 39–48, 2000.

[3] J. Abello, J. Korn, and M. Kreuseler. Navigating giga-graphs. In *Proc. of the working conference on advanced visual interfaces (AVI)*, 2002.

[4] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[5] D. Auber, Y. Chiricota, F. Jourdan, and G. Melancon. Multiscale visualization of small world networks. In *Proc. of IEEE Symposium on Information Visualization*, pages 75–81, 2003.

[6] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz. Approximating aggregate queries about Web pages via random walks. In *Proc. of the VLDB Conference*, pages 535–544, September, Cairo 2000. Morgan Kaufmann.

[7] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. In *Proc. of the World Wide Web Conference*, pages 309–320, Amsterdam, May 2000.

[8] S. Chaudhuri, R. Motwani, and V. Narasayya. On random sampling over joins. In *Proc. of the SIGMOD Conference*, pages 263–274. ACM Press, 1999.

[9] A. Crosby. *The Measure of Reality: Quantification in Western Europe, 1250-1600*. Cambridge University Press, 1997. A summary is online at www.stolaf.edu/other/ql/crosby.html.

[10] S. Dill, R. Kumar, K. Mccurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins. Self-similarity in the Web. In *Proc. of the VLDB Conference*, pages 69–78, September 2001.

[11] P. Eades and M. Huang. Navigating clustered graphs using force-directed methods. *Journal of Graph Algorithms and Applications: Special Issue on Selected Papers from 1998 Symp. Graph Drawing*, 4(3):157–181, 2000.

[12] O. Frank. Estimation of the number of connected components in a graph by using a sampled subgraph. *Scandinavian Journal of Statistics*, 5:177–188, 1978.

[13] G. Furnas. Generalized fisheye views. In *Proc. of the Conference on Human Factors in Computing Systems*, pages 16–23. ACM, 1986.

[14] E. Gansner, Y. Koren, and S. North. Topological fisheye views for visualizing large graphs. In *Proc. of the IEEE Symposium on Information Visualization*, 2004.

[15] P. Haas and C. Koenig. A bi-level bernoulli scheme for database sampling. In *Proc. of the SIGMOD Conference*, pages 275–286, Paris, 2004. ACM Press.

[16] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform url sampling. In *Proc. of the World Wide Web Conference*, pages 295–308, Amsterdam, May 2000. Elsevier Science.

[17] I. Herman, M. Marshall, G. Melancon, D. Duke, M. Delest, and J.-P. Domenger. Skeletal images as visual cues in graph visualization. In *Proc. of the Data Visualization*, pages 13–29, 1999.

[18] J. Huotari, K. Lyytinen, and M. Niemel. Improving graphical information system model use with elision and connecting lines. *ACM Transactions on Computer-Human Interaction*, 11(1):26–58, March 2004.

[19] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[20] A. Klovdahl. A note on images of networks. *Social Networks*, 3:197–214, 1981.

[21] V. Krishnamurthy, J. Sun, M. Faloutsos, and S. Tauro. Sampling Internet topologies: how small can we go? In *Proc. of International Conference on Internet Computing*, Las Vegas, 2003.

[22] LEDA. class library. *www.algorithmic-solutions.com/enleda.htm.*

[23] A. Mendelzon. Visualizing the world wide web. In *Proc. of the working conference on advanced visual interfaces (AVI)*, 1996.

[24] T. Munzner. *Interactive visualization of large graphs and networks.* PhD thesis, Stanford University, 2000.

[25] E. Noik. *Dynamic fisheye views: combining dynamic queries and mapping with database views.* PhD thesis, University of Toronto, 1996.

[26] F. Olken. *Random Sampling from Databases.* PhD thesis, University of California at Berkeley, 1993.

[27] F. Olken and D. Rotem. Random sampling from databases - a survey. *Statistics and Computing*, 5(1):25–42, March 1995.

[28] C. Palmer, P. Gibbons, and C. Faloutsos. Data mining on large graphs. In *Proc. ACM Intl. Conf. on SIGKDD*, pages 81–90, 2002.

[29] C. Plaisant, J. Grosjean, and B. Bederson. Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proc. of the IEEE Symposium on Information Visualization*, pages 57–66, Boston, October 2002.

[30] S. Thompson. *Sampling.* Wiley, 2nd edition, 2002.

[31] D. Watts. *Small worlds: the dynamics of networks between order and randomness.* Princeton University Press, 1999.

APPENDIX

A. A RANDOM ORDERING OF THE EDGES IN THE RUNNING EXAMPLE OF SECTION 4

Our examples in Section 4 use the following random ordering of the edges shown in Figure 6: (b1,b10), (a1,a5), (c1,c3), (b1,b9), (d1,d2), (d3,d4), (a7,b11), (c2,c6), (c11,c7), (c11,c6), (a1,a7), (a5,a13), (c3,c8), (b1,b7), (a5,a14), (c3,c7), (a5,a12), (a7,a16), (a2,a9), (c1,c4), (c2,c9), (c4,c9), (d4,d1), (a2,a10), (c2,c8), (b1,b4), (c1,c5), (a1,a4), (b1,b3), (a4,a11), (b1,b5), (c3,c6), (a1,a2), (d2,d3), (c1,c2), (b1,b6), (c2,c7), (c11,c9), (c5,c9), (a1,a3), (b1,b11), (a7,a15), (c11,c10), (a1,a6), (b1,b8), (c4,c6), (e2,e3), (e1,e2), (c4,c8), (b1,b2), (c11,c8), (c5,c10), (a2,a8), (c4,c7), (c3,c9).

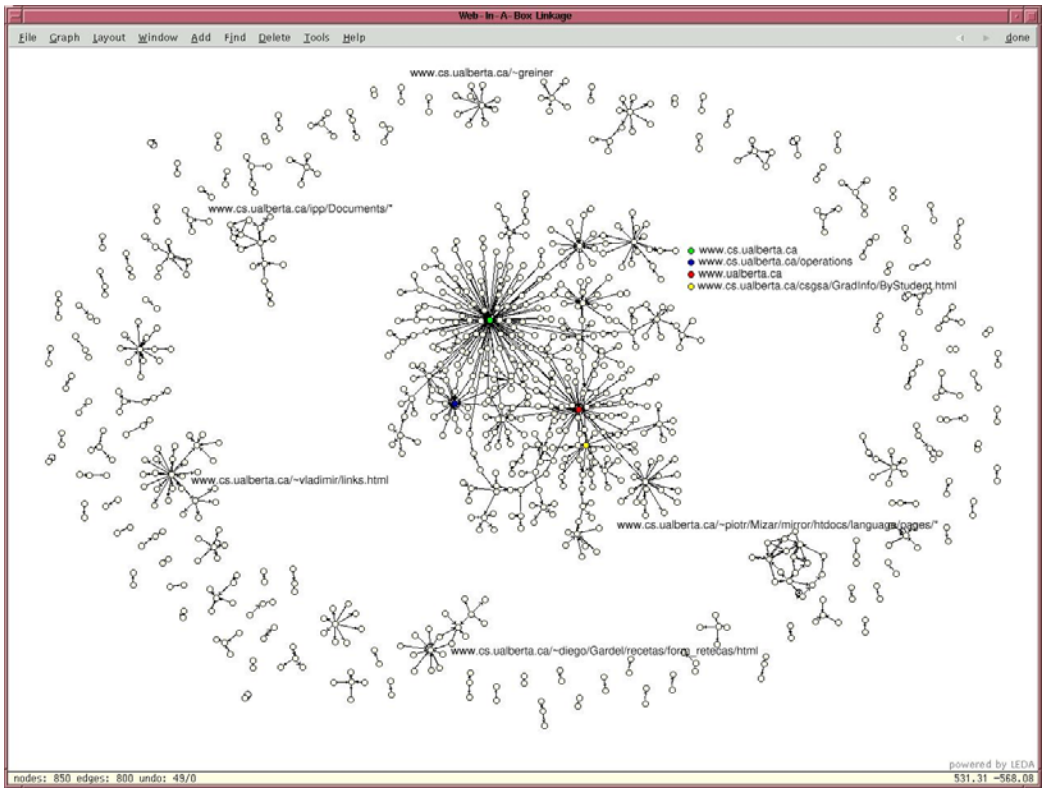


Figure 9: *Interconnections of pages at www.cs.ualberta.ca with the rest of the Web.*

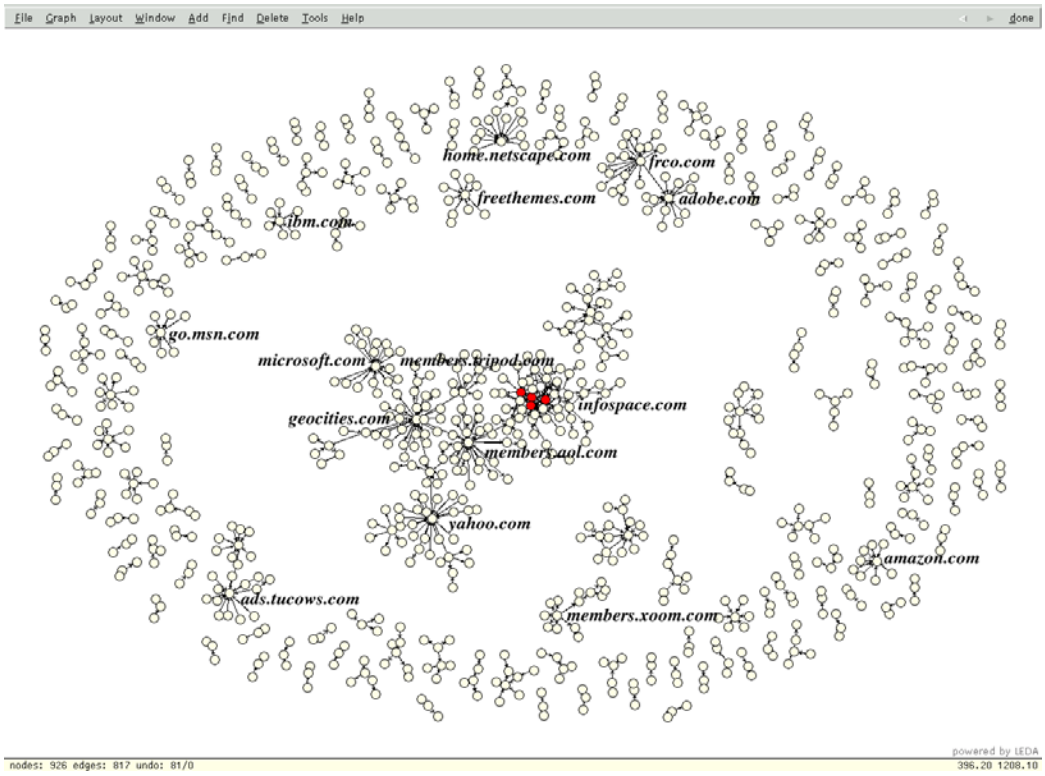


Figure 10: *A random sample of the Web graph at the site level.*

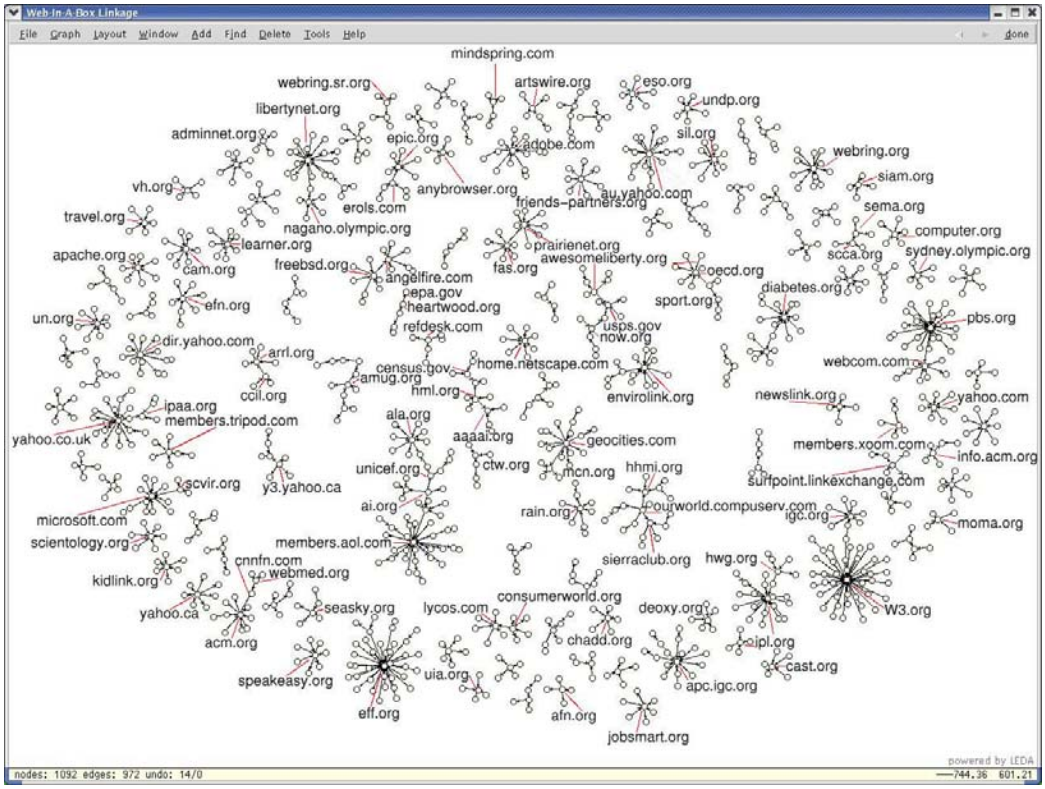


Figure 11: *A focused growth at rate (1,0) of sites in the "org" domain.*

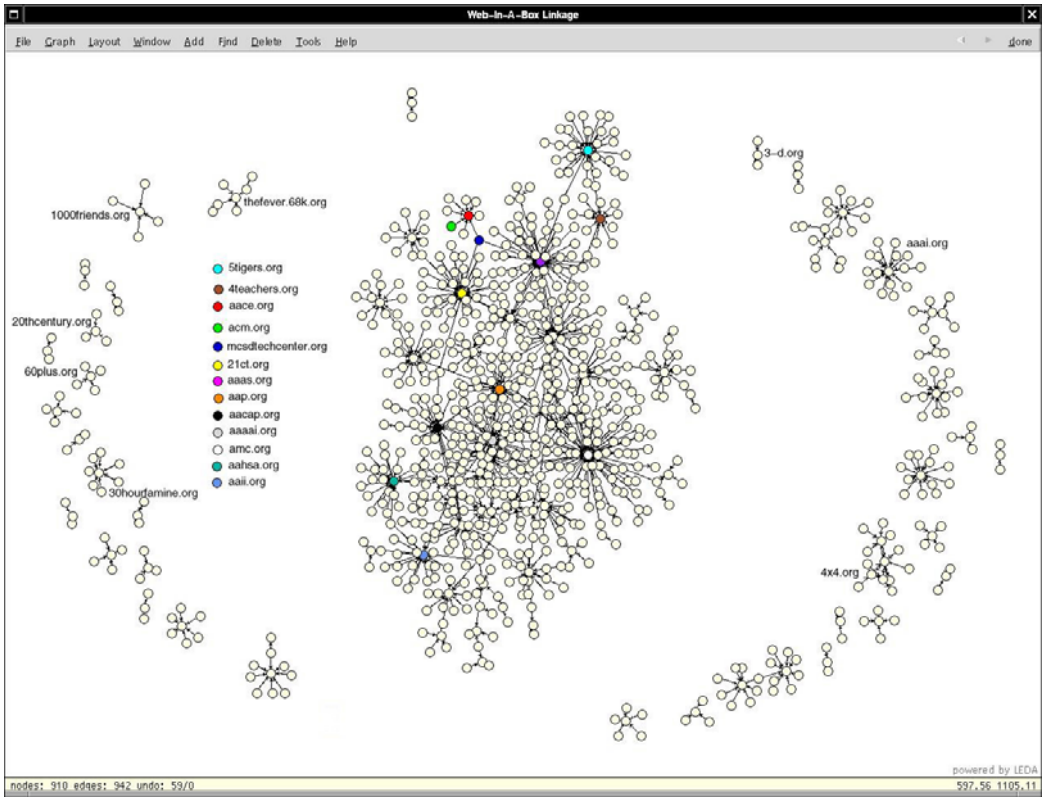


Figure 12: *A focused growth at rate (1,1) of sites in the "org" domain.*

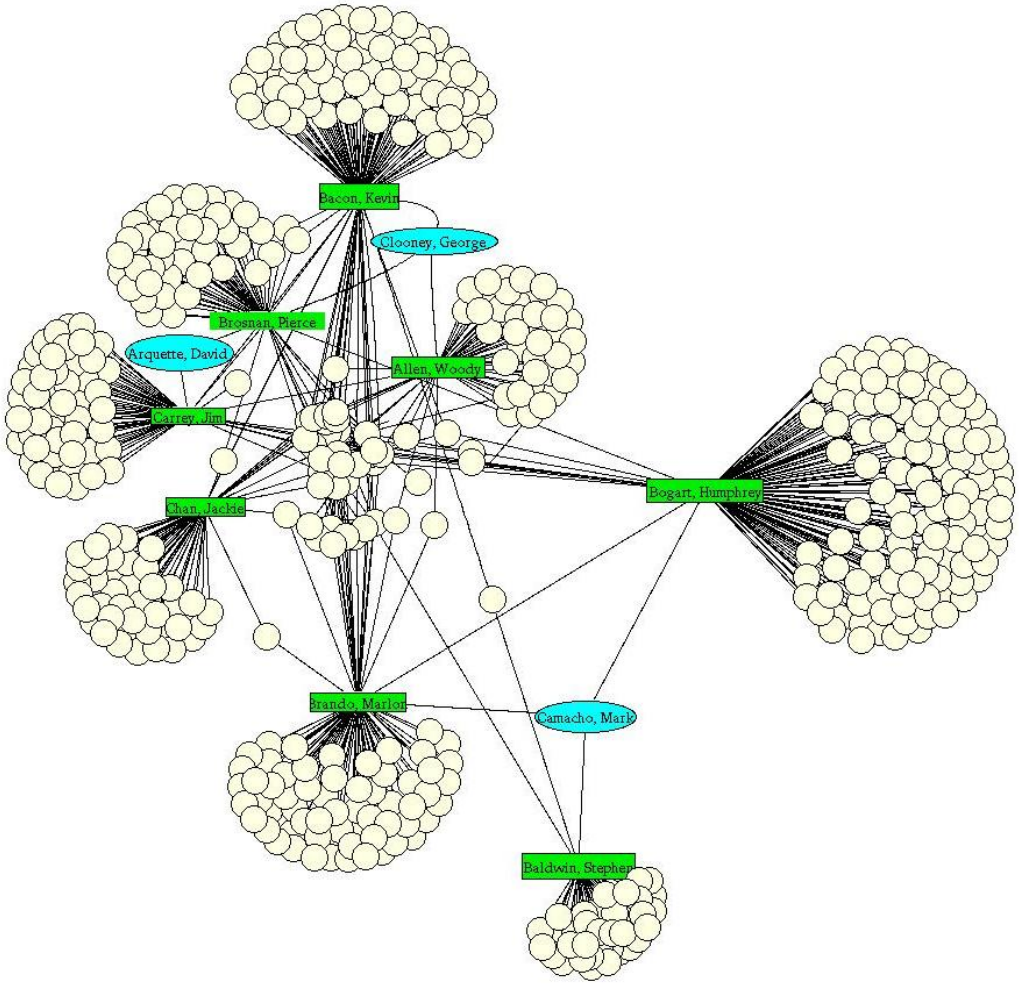


Figure 13: *A mixed growth of the movie database.*