Deliverable D1.2.2

Extensive crawling of the Web and preliminary analysis of the collected data.

Development of data stream and random sampling algorithms for massive Webgraphs.

Interplay between analysis and visualization of dynamic networks.
Start date of the project: January 2004
Duration: 48 months
Project Coordinator: Prof. Dr. math. Friedhelm Meyer auf der Heide
Heinz Nixdorf Institute, University of Paderborn, Germany
Due date of deliverable: December 2005
Actual submission date: January 2006
Dissemination level: PU – public

Work Package 1.2: Monitoring and Measuring the Evolving Web
Participants: Università Degli Studi di Roma La Sapienza (UDRLS), Italy
Universität Paderborn (UPB), Germany
Universität Karlsruhe (UniKar), Germany
Université de Paris Sud (UPS), France
Università Degli Studi di Milano (UniMI), Italy

Authors of deliverable: Stefano Leonardi (Stefano.Leonardi@dis.uniroma1.it)
Alain Barrat (Alain.Barrat@th.u-psud.fr)
Luciana Salete Buriol (Buriol@dis.uniroma1.it)
Debora Donato (Donato@dis.uniroma1.it)
Marco Gaertler (gaertler@ira.uka.de)
Christian Sohler (sohler@uni-paderborn.de)
1 Introduction

The evolving nature of the Web is the main issue on which DELIS Work Package 1.2 is focused. The activity of monitoring the dynamic features of complex network requires the definition of concepts and methods that allow to reconstruct a picture of the Web at a specific time. The fields of application go beyond Web analysis. Dynamic networks naturally occur in many applications such as optimizing transportation, offering and maintaining technical services or ensuring global communication. In most cases a dynamic network is given by a sequence of networks. Each network is a snapshot of an ongoing process. Such a snapshot contains all elements present at a given time. Consider the example of the Internet, where packages have to be routed through physical links. There, changes can alter the cost, the technical availability or the commercial usability of edges (and nodes).

The analysis of dynamic networks can roughly be classified into three parts:

1. the identification of baseline features of the dynamics is a fundamental issue;
2. the analysis of the deviations from this behavior on a microscopic and macroscopic scale;
3. the prediction of the future evolution.

The first issue involves the design of methods that allow to associate vertices and hyperlinks with temporal information. The framework obtained in this way constitutes the starting point for a study of the the evolution over time of the main statistical properties of the Webgraph, as for instance degree-distribution, connectivity properties, spectral properties, quality of pages, the topical regions of the Web. It is worth to underline that extracting the link structure of the Web at a certain point in time is not simple. An attempt can be made by collecting a series of static snapshots by sequential crawls. From the analysis of these snapshots, it can be inferred if a page has been modified or deleted during a certain time frame but it is not possible to determine exactly the instant when the update or deletion occurred. To overcome this problem we study the hyperlinked graphs, wikigraphs, originated from the link structure of the pages of the online encyclopedia Wikipedia. Associated with each node there are timestamps, indicating the creation and update dates of each page, that allows to study how the graph properties evolve over time. Our study reveals that wikigraphs maintain the main characteristics of webgraphs, for which temporal information is usually not available. We also analyze the temporal evolution of several topological properties of wikigraphs and relate this measures to the number of updates of the documents.

It is well known that the computation of statistical properties and network indices is often unfeasible on large networks. This problem is even accentuated in a dynamical environment where each network is seen as a sequence of snapshots. A natural way to address the problem of computing with massive data sets is to resort to algorithms conceived to work in the data stream model [HRR98, Mut05]. We present data stream algorithms which compute a $(1 + \epsilon)$-approximation of the number of certain subgraphs in a graph with probability $1 - \delta$. In this work we specifically present unbiased estimators for the number of triangles in the graph and the number of cliques of any size. We also provide an optimized implementation of our data stream algorithms and test it on networks of various size collected in different application domains.

Visualization can support the first two issues. For example, layouts of different snapshots over time, which are based on the same visualization paradigms, can be utilized to identify stability and similar phenomena. More precisely, the visualization can point out a range of prominent features that might lead to promising results. Thus, it reduces the number of applied measurements neccessary to grasp the dynamic process. We continue our efforts to visualize large and complex networks focusing on the investigation of analytic and dynamic properties. Inspired by structural phenomena observed in the Autonomous System network, we devise alternative means of visualization. As a key feature we direct our attention to the trade-off between achieved quality and required computation time.
Currently, we are experimenting with pseudo-abstract layouts, i.e., layouts where the global shape is predefined by an analysis.

Moreover we propose a visualization algorithm based on the \( k \)-core decomposition able to uncover in a two-dimensional layout several topological and hierarchical properties of large scale networks. The \( k \)-core decomposition consists in identifying particular subsets of the graph, called \( k \)-cores, each one obtained by recursively removing all the vertices of degree smaller than \( k \), until the degree of all remaining vertices is larger than or equal to \( k \). Larger values of \( k \) clearly correspond to vertices with larger degree and more central position in the network’s structure.

2 State of the art

Part of the work performed within the WP 1.2 aims to verify if any evolving trend is observable in the statistical properties of the Web (degree, PR, number of updates) and if these measures are correlated each other over the time. We want to stress that, up to now, very little research work has been devoted to the evolution of the statistical and topological properties of hyperlinked graph as webgraphs, blog graphs and wikigraphs even if the study of the temporal evolution of webgraphs has already been addressed in several previous works [ACH’01, BYBKT04, CGM00, FMNW03, NCO04].

Most of them traced a set of pages in order to compile some statistics about the frequency and rate of the changed pages and the percentage of pages that are deleted or created every year. The search engine perspective is dominant in all of them.

The paper [CGM00] presents the results of an experiment conducted over 4 months. The authors daily crawled 270 sites in order to measure the rate of change and the lifespan of each page. A Poisson process was used to model the rate of change and compare the efficiency of different crawling strategies. The authors also described the architecture of a incremental crawler able to keep up the index with the evolving web.

Fetterly et al. [FMNW03] expanded the work of [CGM00] both in terms of coverage and sensitivity to changes. They found out that good predictors of future changes in the web are the top-level domain pages, and relate document size and history to the freshness of a web page collection.

A search engine-centric approach is followed also in [NCO04]. The authors crawled 154 ‘popular’ sites for a year and revealed a high dynamical behavior of the Web. But, despite of the high rate of newly created pages, the ‘new contents’ introduced are less than 5% of all changes introduced. They also observed that the Web link structure is even more dynamic with more than 75% of new links every year. Moreover they found out that, for pages with significant changes over the time, the degree of changes tends to be highly predictable and observed that this results can be used to crawl proper portions of the Web.

Models for analyzing the evolution of the webgraph were presented in [KHS03, KNRT03]. In particular Kraft et al. [KHS03] defined the notion of TimeLinks and extract some statistics over the data. Kumar et al. [KNRT03] introduced the notion of time graph and conducted a series of experiments in order to trace the formation and the development of communities in the Blogspace and to detect burst of activity within them.

Despite the many applications of counting frequent patterns in graphs, the current state of the art only provides methods that are either computational unfeasible on large networks or do not offer any guarantee on the accuracy of the estimation. In the data stream model data arrives in a stream, one item at a time, and the algorithms are required to use very little space and per-item processing time. Data stream computation allows to summarize data and compute on-line relevant quantities without incurring a large cost for organizing and storing data. Bar-Yossef, Kumar and Sivakumar [ZBY02] give a first solution for counting triangles in the data stream model. They consider both the “adjacency stream model” where the graph is presented as a sequence of edges in arbitrary
order, and the "incidence stream" model where they consider only bounded-degree graphs and all edges incident to a vertex are presented successively. These algorithms are obtained through a so-called "list" efficient reduction to the problem of computing frequency moments [AMS96]. For the adjacency stream model, more algorithms have also been developed in a more recent work [JG05]. These solutions are still far from being practical for most real world networks. We continue our efforts to visualize large and complex networks focusing on the investigation of analytic and dynamic properties. Inspired by structural phenomena observed in the Autonomous System network, we devise alternative means of visualization. As a key feature we direct our attention to the trade-off between achieved quality and required computation time. Currently, we are experimenting with pseudo-abstract layouts, i.e., layouts where the global shape is predefined by an analysis.

3 Main contributions

3.1 Link and Temporal Analysis of Wikigraphs

Wikipedia has more than 1 million articles, available in more than 100 languages. For our goals we generated six wikigraphs, wikiEN, wikiDE, wikiFR, wikiES, wikiIT and wikiPT, from the English, German, French, Spanish, Italian and Portuguese datasets, respectively.

This study reveals that the wikigraphs share the same properties of the samples collected crawling large portions of the Web and that they are characterized by a well-connected structure with more than 80% of the nodes belonging to a large strongly connected component. The temporal analysis shows that

- the single snapshots exhibit very similar properties with respect to indegree and outdegree distribution;
- the number of updates of pages is concentrated in the few first months of its creation.
- the number of updates is not correlated with pagerank, indegree, outdegree and number of visits.

3.1.1 Link Analysis of Wikigraphs

As a very first part of this work, we show the results of the same static analysis outlined in [BKM+00] by Broder et al. that we performed in order to emphasize the similarities between the common webgraphs and the wikigraphs. Consequently we carried out different kind of measurements. We present:

1. the distribution of local measures as indegree and outdegree;
2. the size of the bow-tie components;
3. the Pagerank values for all the pages.

The first set of measurements concerns the indegree and outdegree distributions. The distribution follows a power law\(^1\). We found \(\gamma = 2.1\), as it has been observed in the indegree distribution of webgraphs [BKM+00, DLLM04]. The outdegree distribution, shown by figure ??, also follows a power law with \(\gamma = 2.4\).

The macroscopic connectivity structure of wikipedia has been characterized by mapping the strongly connected components of wikigraphs. A similar analysis applied to the webgraph has revealed a so-called bow-tie structure [BKM+00]. We measure the size of each component of the bow-tie and the results are presented in table 1.

\(^{1}\)i.e. the probability that a node has in-degree \(i\) is proportional to \(\frac{1}{\gamma^i}\), for \(\gamma > 1\)
We can observe that the components sizes differ quite a bit from the previous measures of webgraphs. About 30% of the nodes are in the core of webgraphs [BKM+00, DLLM04], while the core of the different wikigraphs contains a percentage of the nodes of the graph that ranges between 67% and 82%.

Table 1: Size of the bow-tie components of the wikigraphs. Each entry in the table presents the percentage of nodes of the corresponding wikigraph that belong to the indicated bow-tie component.

<table>
<thead>
<tr>
<th>DB</th>
<th>SCC</th>
<th>IN</th>
<th>OUT</th>
<th>TENDRILS</th>
<th>TUBES</th>
<th>DISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT</td>
<td>67.14</td>
<td>6.79</td>
<td>15.85</td>
<td>1.65</td>
<td>0.03</td>
<td>7.50</td>
</tr>
<tr>
<td>IT</td>
<td>82.76</td>
<td>6.83</td>
<td>6.81</td>
<td>0.52</td>
<td>0.00</td>
<td>3.10</td>
</tr>
<tr>
<td>ES</td>
<td>71.86</td>
<td>12.01</td>
<td>8.15</td>
<td>2.76</td>
<td>0.07</td>
<td>6.34</td>
</tr>
<tr>
<td>FR</td>
<td>82.57</td>
<td>6.12</td>
<td>7.89</td>
<td>0.38</td>
<td>0.00</td>
<td>3.04</td>
</tr>
<tr>
<td>DE</td>
<td>89.05</td>
<td>5.61</td>
<td>3.95</td>
<td>0.10</td>
<td>0.00</td>
<td>1.29</td>
</tr>
<tr>
<td>EN</td>
<td>82.41</td>
<td>6.63</td>
<td>6.73</td>
<td>0.57</td>
<td>0.02</td>
<td>3.65</td>
</tr>
</tbody>
</table>

Hence we can conclude that the link structure of Wikipedia is well interconnected, in the sense that most of the nodes are in the core, and from any page it is possible to reach almost any other. Not surprisingly, this is probably due to the implicit aim of an online encyclopedia, that is driving the reader to related topics on the same encyclopedia during the topic description. In this way the content of each article can be fully understood while the surfer visits many articles.

We complete our link analysis by measuring the pagerank distribution for wikiEN. It is a power law function with $\gamma = 2.1$. Previous measures for the webgraphs [DLLM04, PRU02] also exhibit the same behavior for the pagerank distribution. Table 2 presents a list of the top pages considering the pagerank values, with the indegree (in), outdegree (out) and number of visits (#vst) of the pages. The indegree of such pages is high, as expected, while the outdegree is variable. We list the number of visits of the top ranked pages to show that this value is surprisingly not related with the pagerank values. This results was also presented in [Syd05], in which very little correlation was found between the link analysis characteristics and the actual number of visits. We have to wonder if this finding is motivated by the nature of an encyclopedia or if it is a common feature of the Web. Indeed, since in Wikipedia, each item concerns a single topic and each topic is the content of exactly one article, it is not clear what we are going to measure with an algorithm like Pagerank: each page is necessarily authoritative since it is the only one related with a particular issue. Instead, if this is a feature of the Web, then we might conclude that Pagerank could not be a so good measure of the authoritativeness for the final users since they prefer to visit pages with lower Pagerank values.

All the properties presented in this section, indegree and outdegree distributions, bow-tie measures and pagerank distribution, are similar on different datasets. Moreover, similar properties are also found in different snapshots of the same datasets. Some of the properties, mainly the pagerank distribution, are not so explicit when considering small graphs. For example, the pagerank of the graph wikiPT does not have a clear visible power law as it is found for the wikiEN.

3.1.2 Temporal Analysis of wikigraphs

In this section we present the temporal analysis of wikigraphs. By temporal analysis we mean the measures that are related with the evolution of the graph over time. The analysis in these section aims to present measures about the frequency of page update and the distribution of the updates along the time life of a page.
Table 2: The ten top ranked pages and their corresponding indegree, outdegree and number of visits.

<table>
<thead>
<tr>
<th>Page title</th>
<th>in</th>
<th>out</th>
<th>#vst</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>25410</td>
<td>737</td>
<td>7</td>
</tr>
<tr>
<td>2000</td>
<td>46992</td>
<td>383</td>
<td>2</td>
</tr>
<tr>
<td>Asia</td>
<td>40807</td>
<td>97</td>
<td>6</td>
</tr>
<tr>
<td>Native_America</td>
<td>39648</td>
<td>439</td>
<td>1</td>
</tr>
<tr>
<td>Hispanic</td>
<td>38638</td>
<td>46</td>
<td>2</td>
</tr>
<tr>
<td>Latino</td>
<td>38518</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>African_American</td>
<td>39020</td>
<td>74</td>
<td>2</td>
</tr>
<tr>
<td>United_States_Census_Bureau</td>
<td>35001</td>
<td>37</td>
<td>2</td>
</tr>
<tr>
<td>Asian</td>
<td>38633</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>France</td>
<td>13253</td>
<td>517</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 1: Temporal indegree distribution for the wikigraph wikiEN.

We start the temporal analysis of wikigraphs by plotting the indegree, outdegree and pagerank distributions for temporal snapshots of wikigraphs. The Figures from 1 to 3 present these plots for the wikiEN dataset. For clarity, the figures present four plots per year, instead of once a month.

We proceed the temporal analysis of wikigraphs by plotting the distribution of the number of the page updates. Figure 4 presents the distribution of number of nodes by the number of updates for the six wikigraphs used for the tests. Each point presents the number of nodes (y axis) that were updated exactly \( x \) times.

This distribution is a power law with \( \gamma = 1.9 \) for the three larger datasets, and a slight different \( \gamma \) for the other three. A power law also characterizes the distribution of the page updates when we concentrate on single snapshots. In this case the value of \( \gamma \) depends on how close to the dataset creation each snapshot is taken. An example of this is shown in figure 5 for four snapshots of the graph wikiFR. Each point represents the number of nodes (y axis) that were updated exactly the corresponding number of times (x axis). The legend indicates the time the snapshot has been considered. We draw a second experiment that aims to give an indication of the distribution of the updates over the time life of the pages. Consider the set of pages \( \mathcal{U} \) created in a given period. If we
fix a percentage $p$, we can plot which is the percentage of the number of nodes (in that specific period of time) that had at least $p$ of their updates done at some specific time after the creation. Figure 6 presents the plots considering $p_1 = \epsilon\%$, $p_2 = 20\%$, $p_3 = 40\%$, $p_4 = 60\%$, $p_5 = 80\%$ and $p_6 = 100\%$. We use $0 < \text{epsilon} \ll 1$ (red line) such that it plots the time when the first update was executed. The data is from the graph wikiEN.

Many pages, once created, are never updated, or just updated a few times. About 20\% of the pages are never updated, as we can see from the plot on Feb/03 for $p_6 = 100\%$. About 20\% of the pages are fully updated in their first month of existence. It could be the case that many pages are updated soon after they are created, but the reason in fact is that many pages are never updated. By the plot $p_1 = \epsilon\%$, we outline that many pages have an update soon after they are created. For example, 70\% of the pages received the first update in their first month of life.

One further experiment was to measure the correlation between the number of updates of a page and the pagerank, indegree, outdegree and number of visits of a page. From this experiment we concluded that there is no correlation between the number of updates and any one of the other measures. Motivated by the previous experiments, we considered the sequence of updates for the five most frequently updated pages and for the five pages with higher pagerank. We observe that
the frequency of updates increases with time and show several peaks. The peaks seem to be less predictable in the pagerank plots than in the most updated pages.

3.2 Counting Triangles in Data Stream

3.2.1 Streams of edges in arbitrary order

We consider undirected graphs without self-loops. Each edge is an unordered pair of nodes \((v, w)\) such that \((v, w) = (w, v)\). We assume that \(V = \{1, \ldots, n\}\) and \(n\) is known in advance. We have access to a stream consisting of all edges in the graph. The edges appear in arbitrary order and no edge is repeated in the stream. There is no bound on the degree of the nodes.

The sample is an arbitrary edge and a node. A random edge is selected choosing the first edge as a sample edge and replacing this edge by the \(i\)th edge of the stream with probability \(1/i\). It may happen that we sample an edge \(e = (a, b)\) of the stream together with a node \(v\), but we do not see the edge \((a, v)\) or \((b, v)\) in the subsequent stream (because they appeared before the edge \(e\)). In this case, we do not detect \(a, b, v\) as a triangle. However, we detect \(a, b, v\), if \((a, b)\) is the first edge of the triangle that appears in the stream. This changes the expected value of \(\beta\) by a factor of 3.
SampleTriangleOnePass

\[ i \leftarrow 1 \]
\[ \text{for each edge } e = (u, w) \text{ in the stream do} \]
\[ \quad \text{Flip a coin. With probability } 1/i \text{ do} \]
\[ \quad \quad a \leftarrow u; \ b \leftarrow w; \]
\[ \quad \quad v \leftarrow \text{Node uniformly chosen from } V \setminus \{a, b\} \]
\[ \quad \quad x \leftarrow \text{false}; \ y \leftarrow \text{false} \]
\[ \quad \text{end do} \]
\[ \quad \text{if } e = (a, v) \text{ then } x \leftarrow \text{true} \]
\[ \quad \text{if } e = (b, v) \text{ then } y \leftarrow \text{true} \]
\[ \quad i \leftarrow i + 1 \]
\[ \text{end for} \]
\[ \text{if } x = \text{true \& \& } y = \text{true then return } \beta \leftarrow 1 \text{ else return } \beta \leftarrow 0. \]

Lemma 3.1 The streaming algorithm SampleTriangleOnePass outputs a value \( \beta \) having expected value

\[ \mathbb{E}[\beta] = \frac{|T_3|}{|T_1| + 2 \cdot |T_2| + 3 \cdot |T_3|}. \]

To implement the method efficiently we use the reservoir sampling algorithm from [Vit85] to select the edge. The selection then uses \( O(\log |V|) \) expected time for each instance of SampleTriangleOnePass. Additionally we use the hash table approach from the previous chapter to efficiently find instances of SampleTriangleOnePass which search for an edge in the stream. Alltogether we get expected \( O(1 + s \cdot \frac{\log |E|}{|E|}) \) update time per edge in the stream.

Theorem 1 There is a 1-Pass streaming algorithm to count the number of triangles in a stream of edges up to a multiplicative error of \( 1 \pm \epsilon \) with probability at least \( 1 - \delta \), which needs \( O(s) \) memory cells and expected update time \( O(1 + s \cdot \frac{\log |E|}{|E|}) \), where

\[ s \geq \frac{3}{\epsilon^2} \cdot \frac{|T_1| + 2 \cdot |T_2| + 3 \cdot |T_3|}{|T_3|} \cdot \ln\left(\frac{2}{\delta}\right). \]
3.2.2 Incidence streams

In an incidence stream all edges incident to the same vertex appear subsequently in the stream. That is, first arrive all edges incident to vertex $v_1$, followed by all edges incident to $v_2$, and so on. The ordering $v_1, \ldots, v_n$ of the vertices can be arbitrarily, i.e. determined by an adversary. We consider undirected graphs and so each edge appears twice (within the incidence list of both incident nodes). There is no bound on the degree of the nodes (in contrast to [ZBY02]).

A sample in this case is a path of length 2, e.g. a node with two adjacent arcs. Instead of counting in advance the number $P$ of paths of length 2 in the graph, we will start an instance of the streaming algorithm for each guess $\tilde{P}$ of the number of length-2-paths in the set $\{1, 2, 4, 8, \ldots, |V|^3\}$. In parallel we will count $P$. At the end we can find one instance started with a value $\tilde{P}$ satisfying $P \leq \tilde{P} < 2P$. We choose the result of this instance as the result of our algorithm.

It can happen that this instance does not get a sample node at all (because the estimated number of nodes was too high). But since the estimation of the number of nodes is at most twice the real value, this failure happens with probability at most $1/2$ and will be detected. When we in parallel run $\log(1/\delta)$ instances of our algorithm, with probability $1 - \delta$ at least one of these algorithms will succeed.

When selecting a sample and verifying if it is part of a triangle, an edge could be missed when the incidence lists of both endnodes appear earlier within the stream (i.e. when the node $v$ is the last of the triangle nodes). Since this happens with probability $1/3$, the expected value of $\beta$ decreases by a factor of $1/3$.

**Lemma 3.2** The streaming algorithm SampleTriangleOnePass2 outputs a value $\beta$ having expected value

$$E[\beta] = \frac{2 \cdot |T_3|}{|T_2| + 3 \cdot |T_3|}$$

**Theorem 2** There is a 1-Pass streaming algorithm to count the number of triangles in incidence streams up to a multiplicative error of $1 \pm \epsilon$ with probability at least $1 - \delta$, which needs $O(s \cdot \log |V|)$ memory cells and amortized expected update time $O(\log(|V|) \cdot (1 + s \cdot (|V|/|E|)))$, where

$$s \geq \frac{3}{\epsilon^2} \cdot \frac{|T_2| + 3 \cdot |T_3|}{|T_3|} \ln \left(\frac{2}{\delta}\right).$$

3.2.3 Counting cliques of arbitrary size

Using the approach of the previous sections we can count cliques of $\alpha$ nodes in incidence streams as well using one pass. Let $S_\alpha$ be the set of $\alpha$-stars ($\alpha$ nodes $v_1, \ldots, v_\alpha$ and edges $(v_1, v_2), \ldots, (v_1, v_\alpha)$) and $K_\alpha$ be the set of cliques of size $\alpha$. Our memory bounds will depend on $|S_\alpha|/|K_\alpha|$. In network analysis we are interested in those network where this ratio is small, for example constant. Because then there is a significantly higher number of cliques of size $\alpha$ among the neighbors of vertices than in a random graph.

3.2.4 Computational experiments

The codes were written in C/C++, and compiled with the gcc compiler. The experiments were performed on a 2.4 GHz Intel Pentium IV computer with 512 MB of RAM. CPU times were measures with the system function getrusage. The time for reading the graphs is not included in the running times that are reported.

The datasets used in the experiments were divided in three subsets, all of them are comprised of real world instances. The graph dimentions varies from 8 thousand nodes up to 135 million nodes, and from 27 thousand arcs up to 1 billion arcs.
The first subset is composed of only one instance named webgraph. This instance is a webgraph of 135 million nodes extracted in 2001 by the WebBase project at Stanford [Zha05]. In this graph, each web page is a node and each hyperlink between web pages is an arc. The second set of instances is composed by instances used in the experiments reported in [SW05], with exception of an instance that is private data and, thus, we do not have it. More details about the instances are available in [SW05]. The third set of instances is originated from the link structure of Wikipedia [DELIS-TR-0254]. Wikipedia is nowadays the largest online encyclopedia, available in more than 100 languages. In these graphs, each article is a node and each hyperlink between nodes identifies a directed arc. A graph is extracted from each language. The experiments were performed considering the graphs wikiEN, wikiDE, wikiFR, wikiES, wikiIT and wikiPT, extracted from the English, German, French, Spanish, Italian and Portuguese languages, respectively.

Short times and good approximation was observed in the results for the one pass algorithm for undirected edges presented in arbitrary order. Better results were found by the one pass algorithm for edges in the incidence list fashion. Table 3 presents results a size of samples of size 10,000, 100,000 and 1,000,000.

For all runs of all instances, considering the three sample sizes presented, and also a sample size of 1,000 (not presented due to space restrictions), always one or more triangles were found in the sample used. The average percentage deviation is very good, even for sample size of 1,000 samples. Considering the absolute values, the average percentage deviation for all instances, but webgraph, are 17.72%, 5.10%, 2.17% and 0.85% for the sample sizes of 1,000, 10,000, 100,000 and 1,000,000, respectively.

We consider that an approximation of 5% is a very good estimative, and so, for this algorithm, a sample set of size 10,000 provides already good results.

For the results in counting the k33’s of a graph, with 1,000,000 at least two runs of each of the instances were successful finding k33’s in the sample set. The number of samples that are k33’s varies considerable for each instance. For example, considering the sample size of 1,000,000, hundreds and thousands of k33’s were found for all instances, but instances from the third set (minus wikiPT), and google-2002. For the other instances, tens of samples are k33’s, and for some just a few, as it is the case for instance wikiEN, that just four and two samples are k33’s in the successful runs presented for it. Running times are very short, for all runs, even when considering sample size of 1,000,000.

### 3.3 Algorithms for the Analysis and Visualization

#### 3.3.1 Interplay between analysis and visualization of dynamic networks

As a first milestone towards dynamic visualizations, we successfully finished our experiments of the promising approach that stacks layouts of successive snapshots transparently on top of each other [GW06]. However, our study clearly indicates that the current technique is only suitable for temporally small or sparse networks. The basis of our evaluations were coauthorship networks that exhibit significant temporal structure. Currently, we lack the user-specific knowledge of features relevant to the perception of the visualization, especially for large or long-evolving networks. Such features include crucial network elements as the focus and exploratory means. In parallel, we experiment with three-dimensional navigation as an interactive means for supporting the visualization. This approach eases the problem of general dynamic visualization by reducing the overall complexity, i.e., one single perspective does not need to exhibit all user-relevant features at once. An interactive exploration relaxes size and visibility constraints by giving the user the freedom to change his point of view.

We developed a new paradigm for drawing complex networks [AHGGW05]. The visualization supports the recognition of abstract features of any given decomposition while drawing all elements. In order to support the visual analysis that focuses on the dependencies of the individual parts.
Table 3: Results for the one pass algorithm for counting triangles in an undirected graph structured as an incidence list. Samples of sizes of 10,000, 100,000 and 1,000,000 were considered.

<table>
<thead>
<tr>
<th>Graph</th>
<th>r=10,000</th>
<th>Time</th>
<th>r=100,000</th>
<th>Time</th>
<th>r=1,000,000</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>webgraph</td>
<td>7,991,057,264</td>
<td>153.78</td>
<td>7,541,370,749</td>
<td>393.78</td>
<td>7,993,479,298</td>
<td>490.56</td>
</tr>
<tr>
<td>actor2004</td>
<td>1,277,161,593</td>
<td>-4.16</td>
<td>1,155,564,261</td>
<td>-1.79</td>
<td>1,181,093,982</td>
<td>0.43</td>
</tr>
<tr>
<td>google-2002</td>
<td>43,935</td>
<td>-1.22</td>
<td>45,489</td>
<td>3.65</td>
<td>44,765</td>
<td>2.00</td>
</tr>
<tr>
<td>authors</td>
<td>45,203</td>
<td>3.20</td>
<td>45,435</td>
<td>3.52</td>
<td>43,704</td>
<td>-0.42</td>
</tr>
<tr>
<td>itdld0304</td>
<td>37,336</td>
<td>-14.91</td>
<td>42,420</td>
<td>7.55</td>
<td>451,481</td>
<td>-0.79</td>
</tr>
<tr>
<td>wikiEN</td>
<td>21,099,884</td>
<td>7.35</td>
<td>20,400,809</td>
<td>5.39</td>
<td>19,905,296</td>
<td>1.44</td>
</tr>
<tr>
<td>wikiDE</td>
<td>17,178,801</td>
<td>-9.87</td>
<td>20,400,809</td>
<td>5.39</td>
<td>19,905,296</td>
<td>1.44</td>
</tr>
<tr>
<td>wikiFR</td>
<td>20,695,192</td>
<td>2.21</td>
<td>20,400,809</td>
<td>5.39</td>
<td>19,905,296</td>
<td>1.44</td>
</tr>
<tr>
<td>wikiES</td>
<td>863,765</td>
<td>4.85</td>
<td>782,798</td>
<td>-1.72</td>
<td>793,282</td>
<td>-0.40</td>
</tr>
<tr>
<td>wikiIT</td>
<td>791,437</td>
<td>-0.63</td>
<td>774,447</td>
<td>-2.76</td>
<td>800,619</td>
<td>0.52</td>
</tr>
<tr>
<td>wikiPT</td>
<td>768,999</td>
<td>-3.45</td>
<td>827,132</td>
<td>3.85</td>
<td>803,774</td>
<td>0.92</td>
</tr>
</tbody>
</table>
| 3.3.2 k-core decomposition: a tool for the visualization of large scale networks

In the reference [DELIS-TR-0187], we present the visualization tool based on k-core decomposition. It is publicly available on the web page http://xavierinformatics.indiana.edu/lanet-vi/, where a guide of the decomposition, we use an annulus as the general underlying shape. This method has been evaluated using real world data and offers surprising readability. More precisely, medium sized networks containing about 10,000 – 100,000 nodes can be laid out in about one minute. Although, the results are quite promising, there is still a lot of potential in improving the trade-off between achieved quality and running time.

Further, we successfully continued the application of analytic visualizations. More precisely, we used such visualizations for the discovery and the exploration of the represented network structure in combination with given analytic measures [AFG+05]. For example, we investigated peer-to-peer networks with respect to the underlying physical Internet in cooperation with TUM (Technische Universität München). Further work includes both the continued analysis of dynamic phenomena and the comparison of different data samples of the Autonomous System network (in cooperation with DIMES).
to the interpretation of the obtained layouts can also be found.

Formally, the $k$-core of a graph $G$ is the connected maximal induced subgraph which has minimum degree greater than or equal to $k$. Roughly speaking, it is the maximal subgraph $H$ of $G$ with the property that the minimum number of edges from any vertex in $H$ towards other vertices of $H$ is at least $k$. Starting from $k = 1$ (for graphs without isolated vertices), a simple recursive algorithm allows to obtain all $k$-cores of a graph.

- Every vertex of a connected graph belongs to the 1-core. In Fig.7, we have highlighted the different core using closed lines of different types. A dashed line encloses all the vertices in the 1-core (the entire graph).

- Then, all vertices of degree $d < 2$ are recursively cut out. In Fig.7 all these vertices are colored in blue. The other vertices maintain a degree $d \geq 2$ also after the pruning of the blue ones, therefore they are not eliminated. The remaining vertices form the 2-core, enclosed by a dotted line.

- Further pruning allows to identify the innermost set of vertices, the 3-core. One can check that all red vertices in Fig.7 have internal degree (i.e. between red vertices) at least 3. This core is highlighted by a dash-dotted line.

![Figure 7: Sketch of the $k$-core decomposition for a small graph. Each closed line contains the set of vertices belonging to a given $k$-core, while colors on the vertices distinguish different $k$-shells.](image)

The use of different colors for the vertices is useful to stress another important property: the shell index of a vertex. A vertex has shell index $k$ if it belongs to the $k$-core but not to the $k + 1$-core. A $k$-shell, collects all vertices with the same value of the shell index, i.e. those vertices that are pruned at the same stage of the procedure. Blue vertices in Fig.7 belong to the 1-shell, green ones to the 2-shell and the red vertices compose the 3-shell that, being the highest core, coincides with the 3-core.

The main features of the layout’s structure listed below are visible in Fig.8 where, for the sake of simplicity, we don’t show any edge. The leftmost panel displays the case in which all $k$-cores have a single component, while in the rightmost one an example of $k$-core fragmentation is reported. Indeed, it is possible that, during the pruning procedure, the remaining nodes forming a $k$-core do not belong to the same connected component. When such a fragmentation occurs, LaNet-vi computes the separated components of the core and displays all of them in a coherent way (see below).

- The **visualization’s layout** is two-dimensional, composed of a series of concentric circular shells (see the five different shells in Fig.8).

- Each **shell** corresponds to a single **shell index** value and all vertices in it are therefore drawn with the **same color**.
Figure 8: The two drawings show the structure of a typical LaNet-vi’s layout in two important cases: on the left, all $k$-cores are connected; on the right, some $k$-cores are composed by more than one connected component. The vertices are arranged in a series of concentric shells, each one corresponding to a particular shell index. The diameter of the shell depends on both the index value and, in case of multiple components (right) also on the relative fraction of vertices belonging to the different components. The color of the vertices corresponds to their index value, while their size is logarithmically proportional to their original degree.

- A **color scale** allows to distinguish different **shell index** values: in LaNet-vi’s images, as in Fig.8, the violet is used for the minimum value $k_{min}$, then nuances of blue, green and yellow compose a graduated scale for higher and higher index values up to the maximum value $k_{max}$ that is colored in red.

- The **diameter** of each $k$-shell depends on the **index** value $k$, and is proportional to $k_{max} - k$. (In Fig.8, the position of each shell is schematized by a circle having the corresponding diameter). The presence of a trivial order relation in the shell index values ensures that all shells are placed in a concentric arrangement. On the other hand, when a $k$-core is fragmented in two or more components, the diameter of the different components depends also on the relative number of vertices belonging to each of them, i.e. the fraction between the number of vertices belonging to that component and the total number of vertices in that shell. This is a very important information, providing a way to distinguish between multiple components at a given shell index value.

- Finally, the **size** of each node is proportional to the **original degree** of that vertex; we use a logarithmic scale for the size of the drawn bullets.

The $k$-core decomposition peels the network layer by layer, revealing the structure of the different shells from the outmost one to the more internal ones. LaNet-vi provides a direct way to distinguish their different hierarchies and structural organization by means of some simple quantities: the radial width of the shells, the presence and size of clusters of vertices in the shells, the correlations between degree and shell index, the distribution of the edges interconnecting vertices of different shells, etc. The following features are useful to extract this structural information out of the laNet-vi visualization.

1) **Shells Width**

In the LaNet-vi’s representations the width can change considerably from shell to shell. The thickness of a shell depends on the shell index properties of the neighbors of the vertices in
Figure 9: Left: Each shell has a certain radial width around its diameter’s values. This width depends on the correlation’s properties of the vertices in the shell. The dashed lines in the figure point out the width of the outmost shell, that corresponds to the lowest shell index value. In the second shell, we have pinpointed two nodes $x$ and $y$. The node $y$ is more internal than $x$ because a larger part of its neighbors belongs to higher index values compared to $x$’s neighbors. Indeed, $y$ has three links to nodes of higher shell index, while $x$ has only one.

Right: The figure shows the clustering properties of nodes in the same shell. In each shell, nodes that are directly connected between them (in the original graph) are drawn close one to the other, as in a cluster. Some of these sets of nodes are circled and highlighted in gray. Three examples of isolated nodes are also indicated; these nodes have no connections with the others of the same shell.

...the corresponding shell. For a given shell-diameter (corresponding to the black circle in the median position of shells in Fig.9), each vertex can be placed more internal or more external with respect to this reference line.

2) **Shell Clusters**

The angular distribution of vertices in the shells is not completely homogeneous. Fig.9 shows that clusters of vertices can be observed. The idea is that of grouping together all nodes of the same shell that are directly linked in the original graph and of representing them close one to another in the shell. Thus, a shell is divided in many angular sectors, each one containing a cluster of vertices. This feature allows to figure out at a glance if the shells are composed of a single large connected component rather than divided into many small clusters, or even if there are isolated vertices (i.e. disconnected from all other nodes in the shell, not from the rest of the $k$-core!).

3) **Degree-Index Correlation**

Another property that can be studied from LaNet-vi’s images is the correlation between the degree of the nodes and the shell index. In fact, both quantities are centrality measures and the presence or the absence of correlations between them is a very important feature characterizing a network’s topology. The nodes displayed in the most internal shells are those forming the central core of the network; the presence of degree-shell index correlations then corresponds to the fact that the central nodes are most likely high-degree hubs of the network. This effect is indeed observed in many real communication networks with a clear hierarchical structure, as the Internet at the Autonomous System level or the World Wide Air-transportation network.
On the contrary, the presence of hubs in external shells is typical of networks without a clear global hierarchical structure as the World-Wide Web or the Internet Router Level. In this case, emerging star-like configurations appear with high degree vertices connected only to very low degree vertices. These vertices are rapidly pruned out in the k-core decomposition even if they have a very high degree, leading to the presence of local hub in the external k-shells. These examples are shown in the Image Gallery of the web site.

4) **Disconnected components**: The fragmentation of any given k-shell in two or more disconnected components is represented by the presence of a corresponding number of circular shells with different centers. The diameter of these circles is related with the number of nodes of each component

In summary, LaNet-vi makes possible a direct, visual investigation of a series of properties:

- hierarchical structures of networks;
- connectivity and clustering properties inside a given shell;
- relations and interconnectivity between different levels of the hierarchy;
- correlations between degree and shell index, i.e. between different measures of centrality.

The use of the $k$-core decomposition is also at the center of another visualization tool: the HalfMoon paradigm [DELIS-TR-0227]. This other tool will soon be as well publicly available.

**References**


