4-2012

Why Is the Shape of the Web a Bowtie?

P. Takis Metaxas
Wellesley College, pmetaxas@wellesley.edu

Why Is the Shape of the Web a Bowtie?

Panagiotis Takis Metaxas
Computer Science Department
Wellesley College
Wellesley, MA02481, USA
Email: pmetaxas@wellesley.edu

***Abstract*—The first time in Graph Theory a graph was characterized as "Bowtie" was in the seminal paper by Broder et. al. Though no textbook had ever mentioned this type of graph before, no less an important network than the Web Graph itself is supposed to resemble this shape. In two large collections of crawled Web pages and in numerous smaller collections, researchers discovered Bowties and Bowtie-looking graphs by studying millions of web pages.**

But why do collections of Web pages resemble a Bowtie? The short answer is "because, given the way the Web is created, that's the only shape it could have". This paper shows why this is the case and presents an algorithm and software to visualize Bowtie graphs.

I. Introduction

The Web is an integral part of our daily lives. Studies report that billions of people are visiting billions of pages on the Web every day [1]. It is reasonable, therefore, that one would like to know what the graph that connects those Web pages looks like. The Web Graph, as it is known, is the graph comprised of Web pages interconnected through the hyperlinks they contain. For simplicity, let's assume that pages on the Web are only "static", that is, they are composed of text and binary files residing on servers on the Internet. This is not always the case, since today's servers can create a "dynamic" page on request, but this is not important in this discussion.

One explores the Web Graph by starting at some page and then visiting other pages by clicking on a hyperlink contained in the currently viewed page. Search engines do the same by "crawling" the Web: downloading the contents of billions of Web pages while following the hyperlinks they encounter in most of them. In either case, however, one can never explore or crawl the whole Web. How can we tell what the Web Graph looks like and why?

Ever since the seminal paper [2], it has been claimed that the Web Graph resembles a "Bowtie". They came to this conclusion by studying the connectivity between millions of web pages, crawled from a few starting points. The name was actually given by Andrei Broder when with his colleagues were trying to make sense of the collected Web data: "Indeed I can personally take full credit (or blame) for the Bowtie moniker – we were drawing all sort of pictures on the boards at SRC in Palo Alto and this shape analogy jumped at me." [3]

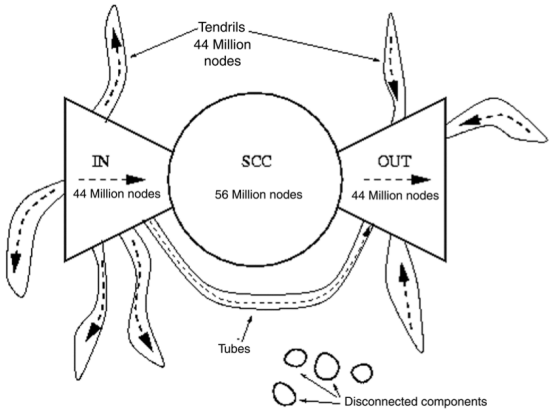

Fig. 1. The shape of the Web Graph (according to [2]) is mainly a Bowtie looking graph with a few extra parts. We will refer to the strongly connected component (SCC) as the CORE, since it more appropriately reflects on its role.

But, given that no one has ever seen the whole Web, how can we be sure? And why does it resemble a Bowtie instead of another shape? A Bowtie, after all, is not among the famous graphs we have studied in the past in Graph Theory. How come did we miss it? Is it a new type of graph, one specific to the Web that could only be encountered there, or is it known by other names in Graph Theory? This paper will try to answer these questions.

The remaining of this paper is as follows: In the next section II we review the various claims on the shape of the Web that researchers have produced in the past, while in section III we explain why Bowtie is the right characterization, and we sketch a proof of this claim. In section IV we present a Web Graph recognition algorithm which can be applied on any directed graph and recognize its Bowtie regions, if there exist. Finally, we end with the conclusions in section V.

II. Claims on the Shape of the Web Graph

The figure that accompanied Broder et. al. [2] paper (see Figure 1) shows a slightly more complicated picture than a simple Bowtie. The directed Web Graph has a strongly connected component in the middle, called the CORE, and two regions of approximately equal size on the two sides of CORE named IN and OUT. There are snake-like regions hanging off IN and OUT, called TENDRILS, and others connecting them without passing through the CORE, called TUBES.

TABLE I
SIZES OF BOWTIE SUBGRAPHS IN TWO STUDIES

	Broder et. al. [2]	Donato et. al. [4]
CORE	28%	33%
IN	21%	11%
OUT	21%	39%
TENDR	22%	13%
ISLAND	9%	4%
Total	203.5 M	200M

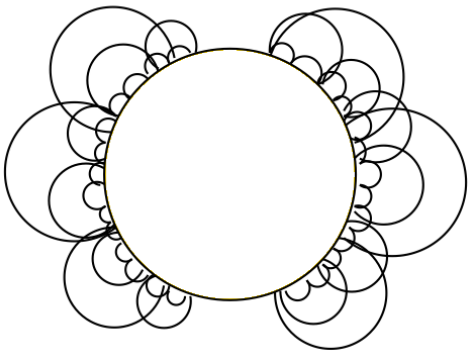

Fig. 2. The shape of the Web Graph (according to [5]) is more accurately represented by a daisy-looking graph.

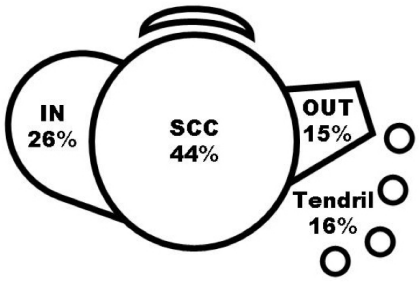

Fig. 3. The shape of the Chinese Web Graph (according to [6]) is a teapot-looking graph.

Fig. 4. As far as we know, no one has claimed that the shape of the Web resembles a cauliflower yet.

Finally, there are several smaller components, represented by circles and disconnected from the large Bowtie. No particular information about their shape of the latter (which we will call ISLANDS [4]) is reported though they may include a tenth of the overall Web size or more.

In a subsequent major study a couple years later ([4]) the sizes of the named regions were estimated to differ significantly from those in the original study, but the Bowtie shape was not disputed. Table I has details on the region sizes in these two major studies.

But is this an accurate picture of the Web Graph? After all, even in 1999 we could only retrieve a small portion of the Web and it is unlikely that anyone tried to draw even a 200 million nodes graph. Bowtie is a conceptual (poetic?) characterization of the Web Graph, something that made sense at the time. Is it possible that, had the authors been able to draw the full Web Graph it would look differently? Or that, ten years later, the Web Graph, probably more than 1,000,000 million nodes strong, looks very differently? Maybe it looks like a daisy [5] (see Figure 2), or a teapot [6] (see Figure 3), or a cauliflower (see Figure 4)? How can we know?

III. THE SHAPE OF THE WEB GRAPH IS A BOWTIE

A main point we make in this paper is this: The shape of the Web is a Bowtie because that's the only likely shape that it could be. Let us explain why this is the case by discussing first what a Bowtie graph is and how it can appear in directed graphs grown by independent distributed processes, such as the millions of Web pages created by people and machines all over the world.

In all of the Web Graph representations, its most important subgraph is its CORE. This is defined to be its largest (not the only) strongly connected component (SCC). All the other named regions are defined in relation to the CORE. In the Bowtie representation, IN is composed of those nodes that are on a directed path that ends on a node in CORE, but that they themselves are not part of the CORE. Similarly, OUT is composed of those nodes that are on a directed path that starts from a node in CORE, but that they themselves are not part of the CORE.

These three major regions are not the only ones present in the Web Graph. We still have not accounted for about a third of the nodes, according to the studies. ISLANDS are nodes completely disconnected from CORE, IN and OUT, that is, there is no directed path that connects them to the Bowtie. They are depicted by generic circles and no indication exists about any internal shape they may resemble. (We will see that they are very likely shaped as Bowties themselves.)

Finally, TENDRILS come in three flavors: TENDRILS-IN are nodes for which there is a directed path from IN, but there is no directed path from them to any other component. Similarly, TENDRILS-OUT are nodes that are on a directed path to a node in OUT, but no path leads from them to any

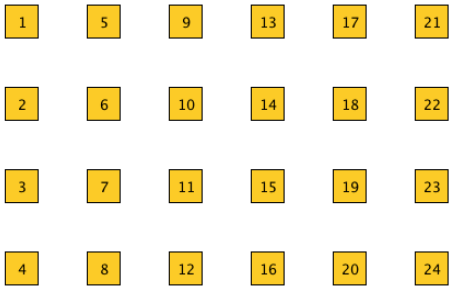

Fig. 5. A collection of websites without hyperlinks.

other region. TUBES are nodes that are on a path from a node in IN to a node in OUT, and there is no path that connects them to CORE. If their connecting paths were to be broken, they would end up as one or more simple TENDRILS.

A. How a Bowtie appears

So, all of the different regions in a Bowtie graph are defined with respect to the CORE. Clearly, if there is no CORE, there is no Bowtie graph. But it is practically impossible for the collection of interconnected web pages in the Web Graph *not* to have an SCC.

Indeed, *given a collection of web pages that are allowed to link one or more times to any other web page of the collection they choose to, an SCC is bound to arise, and with it a Bowtie*.

The above claim is one that can be proven formally, and we will give here the pieces of a constructive proof while keeping the discussion informal. The main pieces of the proof are as follows.

Consider a collection of Web pages that initially are not linking to any other page (Figure 5). These web pages do contain hyperlink references, but we have not consider them yet. We will consider them in three stages.

First Link. Consider that each node introduces a single link to other nodes in the collection. As a result, a pseudo-forest will arise (Figure 6). A pseudo-forest is a collection of pseudo-trees, which in turn is a maximally connected set of N nodes with 1 directed edge (arc) per node. You can think of a pseudotree as an inverted tree with one more arc that is bound to create a directed cycle – thus the prefix "pseudo" in its name.

The existence of pseudotrees guarantees the existence of directed cycles in the graph. The members of each such directed cycle are (by definition) members of a strongly connected component creating the first candidates for the CORE! The largest of those cycles is defined to be the current CORE, though this characterization may change as more links are considered.

Pseudotrees arise often in parallel computation (e.g., see [7]), when a collection of processors try to elect a representative by randomly selecting another processor in the collection. In this case, the pseudotree's cycle must be broken in order to select the representative. But on the Web, the members of such SCCs are gaining some prominence and end up having a greater PageRank [8].

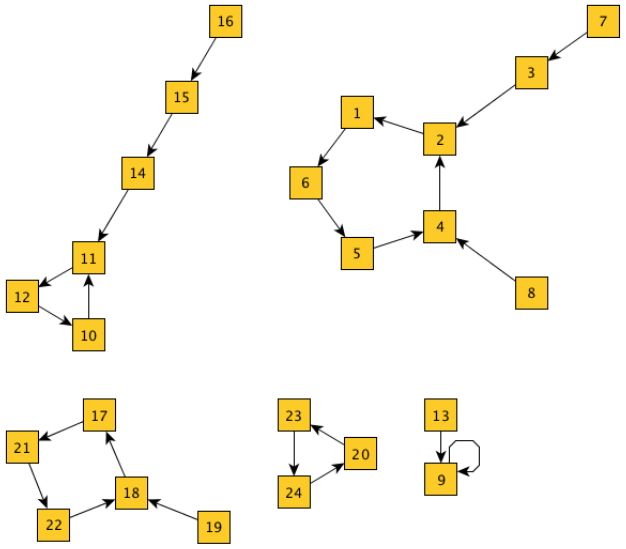

Fig. 6. Each web site has chosen independently one link to a site in the collection. In this instance, five pseudo-trees appear, the smallest one containing two nodes (one with a self-loop)

Note that even with just one link per node, several other components of a Bowtie also appear. The SCCs are likely to have "tails" which, for the CORE, is defined as members of the IN group. However, with just one link per page, OUT or TENDRILS will not appear. This will change with the introduction of a second link.

Pseudotrees with smaller SCCs will be defined as ISLANDs. It is worth noting that these ISLANDs will have similar structure to the Bowtie that contains the CORE, since they are pseudotrees themselves. In other words, the ISLANDs themselves look like Bowties, settling one of the questions we posed above[1]. We can now correct the famous drawing from [2] to reflect this fact. See Figure 7.

Second Link. When a second link per web page is considered, the CORE (and the other SCCs) is likely to increase in size. This happens whenever any member from the current CORE links to a web page in IN, or when a page from OUT links to a page in the CORE or in the IN, to name but three simple cases.

Moreover, the number of ISLANDs is likely to

[1]A related fact is that any sub-collection of nodes in a graph created by crawled collection of nodes is bound to contain Bowties, no matter if this collection is in CORE, IN, OUT, etc. This fact, termed "self-similarity" was studied first in [9]. Our observation explains why this was to be expected.

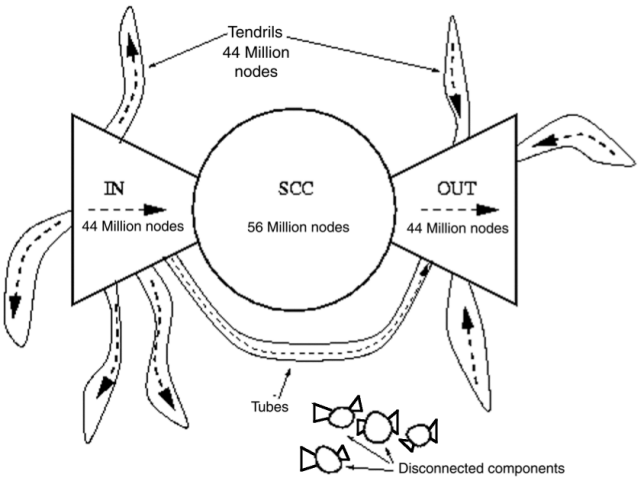

Fig. 7. A more accurate description of the Web is that it is composed of a collection of Bowties, since the disconnected components must also resemble Bowties, in general. In this figure, we have edited the famous figure by [2] to reflect this fact.

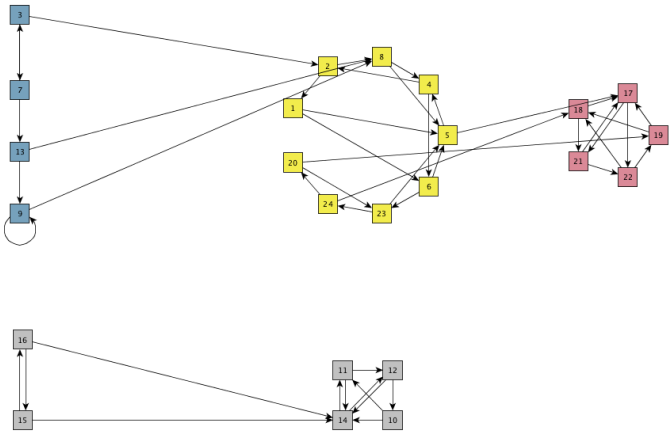

Fig. 8. Each web site has chosen a second link to a site in the collection. A recognizable Bowtie structure has already emerged, along with an ISLAND. The coloring and the placement of the nodes indicate the characterization of the node's region, and is produced by the visualization software we describe in the next section.

decrease, e.g., when a node from an ISLAND gets connected to the Bowtie, the size of the Bowtie increases and the ISLAND disappears (Figure 8): If the connecting link comes from the ISLAND, it becomes part of the IN; if it comes from the CORE or OUT, becomes part of the OUT; if it comes from the IN becomes part of the TENDRILS.
In our construction so far we did not address what happens to web pages that will never link to other pages, such as images, PDF files or other file formats that may contain no links. After considering the introduction of the first link we allow these nodes to participate in the construction by allowing them to be linked by other nodes. These nodes without any links of their own will be part of the OUT or ISLANDs regions.

Remaining Links So, with just two links per node, all the regions that compose the Bowtie are potentially in place. As the third and other links are added, the CORE and the Bowtie will grow even more. For illustrative purposes, Figure 6 shows five connected components of similar size. This is not likely what would happen in practice if nodes were to select other nodes at random. In fact, it takes little over a linear number of random links to weakly connect the whole collection [10].

We should point out that what we present above is not the order in which links were actually added in the real Web. It is a scenario of how links *could have been* added, but likely they were not. This, however, does not affect the recognition of the Bowtie regions, as the links we consider in the third step would only increase the size of the Bowtie regions. This property is useful in the correctness of our construction since, in order to work, we need to start with a collection of nodes that have at least two links to members of the collection – not a hard requirement to satisfy – think of the pages in your favorite site. Of course, if links were deleted at some point, our analysis can completely ignore them and instead focus on any current state of the Web.

One final point of clarification for our recognition model is that the characterization of the CORE may change over time, especially in the early stages. What was CORE may become just an SCC because it may be the case that some ISLANDs got linked in a way that produced a larger CORE. There is no problem in the proof with that because as we explained, ISLANDs are also Bowties.

It may also be the case that another SCC that happened to be outside the CORE region grew so much that its size grew larger than the CORE. It will be named CORE and the characterization of some of the other nodes in the Bowtie may change. For example, in Figure 8, node 22 may select a third link to 16 and node 12 to 22, joining the 5-member SCC in the OUT region, while no structural change may happen in the current SCC. The new 9-member SCC could now be considered the CORE while the old would be part of IN (see Figure 9). So, the characterizations of the Bowtie regions may change without affecting the correctness of the proof about the overall shape of the graph as a Bowtie.

IV. A Bowtie Recognition Algorithm

We now turn out attention to a Bowtie recognition algorithm that, given a directed graph, will categorize each vertex by the Bowtie component that it belongs to. If the input is simply a directed acyclic graph (a "DAG"), it will report that there is no CORE, and therefore no Bowtie will be recognized.

We have also created a visualization program that draws such Bowties using the *yEd* graph drawing package [11]. It takes as input the output of the Bowtie recognition algorithm, colors the nodes according to their region characterization,

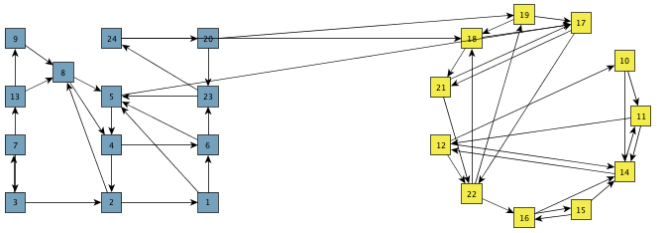

Fig. 9. An example on how the characterization of CORE could change without affecting the existence of a Bowtie graph.

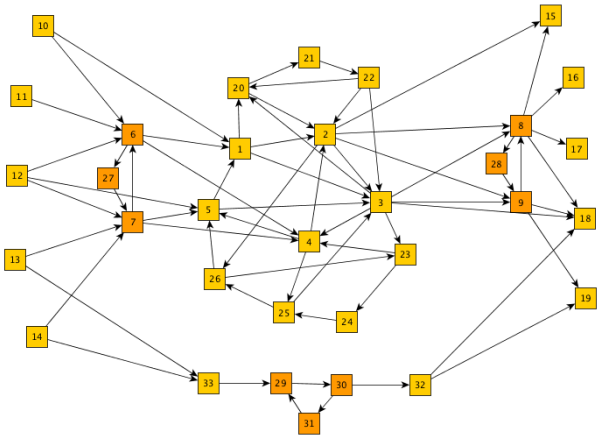

Fig. 10. The input graph to illustrate the Bowtie recognition algorithm. It contains four non-trivial SCCs, one in each of the CORE, IN, OUT and TENDRILS regions, the last three drawn in a different color so that they can be easily identified visually.

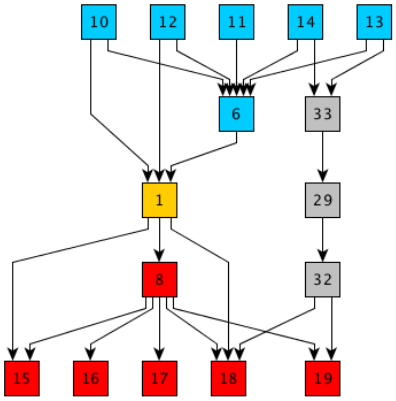

Fig. 11. The SCCs of the input graph in Figure 10 have been identified and contracted in one representative node per SCC. This contracted graph C_i is a directed acyclic graph (DAG).

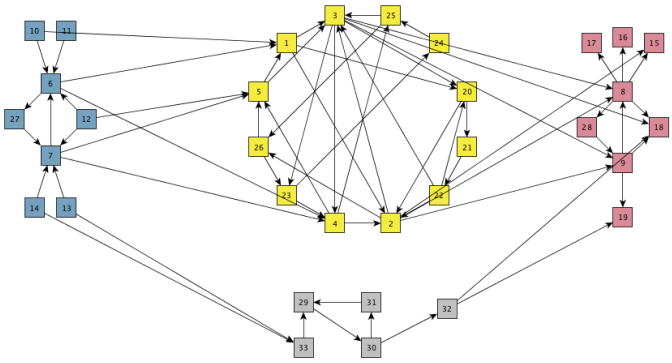

Fig. 12. The output of our algorithm has recognized the Bowtie regions and draw them so that they resemble, well, a Bowtie.

and places the colored nodes in a gml-formatted file for visualization. We will not describe the visualization software in this paper, but we will note that all of the example pictures we have included in this paper were drawn by it.

The first step of the algorithm is to separate the input graph into weakly connected components. To do that we take the undirected version of the input directed graph and recognize its connected components. Each of them is a different weakly connected component that we will process as a separate Bowtie. The one that contains the largest SCC will be the main Bowtie while the rest will be labeled as ISLANDs. Without loss of generality, in the remaining of our description we will assume we only have one weakly connected component, W_i, as the algorithm will treat them all in the same manner.

To facilitate the illustration, we refer to Figure 10 as our input graph. Note that it contains 4 non-trivial SCCs, one in each of the CORE, IN, OUT and TENDRILS regions.

For each of the weakly connected components W_i we do the following:

1) First, we compute the strongly connected components (SCCs) of W_i. To do that, we use the linear-time algorithm described in [12] that involves first a DFS in the input graph and then a DFS of the transpose graph.[2] It will potentially have many strongly connected components, but the largest will be defined to be the $CORE_i$. The remaining ones could be included in the IN_i, OUT_i, or $TENDRILS_i$.
2) Next, we compress the input graph so that each SCC becomes a single node in the compressed graph C_i. (See Figure 11.) This can also be accomplished in linear time.
3) We run a DFS on the compressed C_i starting from the node representing the $CORE_i$. Those nodes reachable from $CORE_i$ are included in OUT_i.
4) Next, we run a DFS on the transpose of C_i starting again from the node representing the $CORE_i$, thus recognizing those vertices to be included in IN_i.
5) Those nodes not characterized at this point are labeled as belonging to $TENDRILS_i$.

Overall, the algorithm runs in linear time, which is not bad for small graphs. The major problem for dealing with huge graphs comes actually from the requirements for space,

[2]The authors of [12] indicate that their algorithm is adapted from Aho, Hopcroft and Ullman, who in turn credit it to S.R. Kosaraju and M. Sharir.

and our implementation can only handle several thousands nodes on a typical laptop. The authors of the large studies we mentioned before were able to deal with larger graphs by sampling random paths in either a pro-processed "connectivity server" database [2], or using external memory algorithms [4].

V. Conclusion

In this paper we presented a proof that the Web Graph resembles a Bowtie – or, more accurately, a collection of Bowties. We also presented an algorithm that, given a directed graph, it will identify the nodes comprising the regions of a Bowtie, if it exists (that is, unless the input graph is not a DAG).

Given the importance of the Web, we expect that the name Bowtie will appear in future Graph Theory textbooks because it is basic and intuitive, a characterization that "worked out because everyone gets it immediately" [3]. The question that is much more difficult to answer is, what are the relative sizes of the Bowtie components of the Web Graph today? That we will likely never know with great accuracy, but we suspect that all Web content provides are trying to make the CORE as large as possible for their own benefit.

In terms of future research, we can use this algorithm to recognize the Bowtie regions of graphs that were generated using particular graph creation strategies. For example, one can use use the algorithm to recognize instances of Bowties from graphs generated by a random process [10] and by processes credited with creating Web Graph-like instances (graphs that share many statistical characteristics of the Web Graph, such as exhibiting powerlaw distribution of in-degree). In the latter category are processes such as Preferential Attachment [13], Copying, and Multi-layer [4]. The interesting aspects of this would be to measure the size of the Bowtie components, and study which parameters may create components that resemble the sizes observed in real samples.

Acknowledgment

This research was partially supported by NSF grant CNS-1117693. The author would like to thank Lorraine Shim for implementing the code that recognizes and visualizes the regions of a Bowtie, and Danaë Metaxa-Kakavouli for reviewing an earlier version of this paper.

References

[1] Pew Foundation, "Pew internet and american life project," http://www.pewinternet.org, 2008.

[2] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," *Comput. Networks*, vol. 33, no. 1-6, pp. 309–320, 2000.

[3] A. Broder, "Personal communication," July 30 2011.

[4] D. Donato, L. Laura, S. Leonardi, and S. Millozzi, "The web as a graph: How far we are," *ACM Trans. Internet Technol.*, vol. 7, February 2007.

[5] D. Donato, S. Leonardi, S. Millozzi, and P. Tsaparas, "Mining the inner structure of the web graph," in *Eigth international workshop on the Web and databases WebDB*, June 2005.

[6] J. J. H. Zhu, T. Meng, Z. Xie, G. Li, and X. Li, "A teapot graph and its hierarchical structure of the chinese web." in *WWW*. ACM, 2008, pp. 1133–1134.

[7] D. B. Johnson and P. T. Metaxas, "Connected components in $O(\log^{3/2} n)$ parallel time for the CREW PRAM," *Journal of Systems Sciences*, vol. 54, no. 2, pp. 227–242, 1997.

[8] M. Bianchini, M. Gori, and F. Scarselli, "Inside pagerank," *ACM Trans. Internet Technol.*, vol. 5, pp. 92–128, February 2005. [Online]. Available: http://doi.acm.org/10.1145/1052934.1052938

[9] S. Dill, R. Kumar, K. S. Mccurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins, "Self-similarity in the web," *ACM Trans. Inter. Tech.*, vol. 2, no. 3, pp. 205–223, 2002.

[10] P. Erdös and A. Rényi, "On the evolution of random graphs," in *Publication Of The Mathematical Institute Of The Hungarian Academy Of Sciences*, 1960, pp. 17–61.

[11] yWorks, "yEd – java graph editor, v. 2.2.1," http://www.yworks.com/en/products_yed_about.htm.

[12] T. T. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*. Cambridge, MA, USA: MIT Press, 1990.

[13] A. Barabási, *Linked: the new science of networks*. Perseus Pub., 2002.