Web-Crawling Reliability

Viv Cothey
School of Computing and Information Technology, University of Wolverhampton, Lichfield Street, Wolverhampton, United Kingdom, WV1 1SB. E-mail: viv.cothey@wlv.ac.uk

In this article, I investigate the reliability, in the social science sense, of collecting informetric data about the World Wide Web by Web crawling. The investigation includes a critical examination of the practice of Web crawling and contrasts the results of *content crawling* with the results of *link crawling*. It is shown that Web crawling by search engines is intentionally biased and selective. I also report the results of a large-scale experimental simulation of Web crawling that illustrates the effects of different crawling policies on data collection. It is concluded that the reliability of Web crawling as a data collection technique is improved by fuller reporting of relevant crawling policies.

Introduction

The Web graph model of the World Wide Web (or just the Web) is now firmly established (Broder et al., 2000). The model provides both an analytic framework for studying the Web and a mental model for discussing the Web (Kleinberg, Kumar, Raghavan, Rajagopalan, & Tomkins, 1999). Equally well established is the principle of operation of Web crawlers (or spiders, robots, wanderers, etc.) (e.g., Gordon & Pathak, 1999; Thelwall, 2001). Bailey, Craswell, and Hawking (2003, p. 4) describe crawling as "the process of identifying and fetching Web pages, usually for indexing, by traversing the Web link graph from a set of starting points." That is, Web crawlers move from node (or document) to node by means of the hyperlinks that each node contains and that define the edges of the Web graph. This Web-crawling technique, coupled with document indexing and retrieval procedures, has proved particularly successful at supporting Web search engine systems. These make relevant Web document content accessible to users. Web crawling is also the data collection technique that supports structural and informetric analyses of the Web graph (e.g., Meghabghab, 2001). In particular, one form of analysis of the Web graph considers the inlinks and outlinks of nodes, which represent document references and citations, respectively (Cronin, 2001; Egghe & Rousseau, 1990).

However, putting the principle of Web crawling into practice raises many design and performance issues (Brin & Page, 1998; Eichmann, 1994; Najork & Heydon, 2001). These give rise to constraints and compromises in operation that affect data collection; the set of operational characteristics of a Web crawler is described here as the Web-crawling policy. This paper investigates the practice of Web crawling and contrasts content crawling with link crawling. *Content crawling* is used here to refer to the objective or goal of the Web-crawling procedure (typically undertaken to support Web search engines) used in conjunction with discovering and indexing the content of documents that compose the Web. Content crawling, for example, ignores duplicate documents and thus differs from link crawling, which does not. In addition, I present the findings of an experiment to discover the effects of different Web-crawling policies to investigate the reliability of Web crawling as a data collection technique.

The notions of reliability and validity that are used here are as commonly understood within the social sciences (for example, Kerlinger & Lee, 2000). That is, the reliability of an investigative technique entails, among other things, whether independent researchers using the technique to investigate the same object of study produce the same result. The validity of an investigative technique considers the extent to which the technique is really describing the phenomenon being reported. Although the two notions are bound together, a technique may yield consistent results that may not be valid. Clearly researchers need access to reliable and valid techniques if their investigations are to build into a coherent and consistent body of knowledge. It will be argued that the reliability of Web crawling is problematic (because the data collected by a Web crawler depend on the crawling policy employed). However, I suggest that this reliability problem, together with issues of validity, can be addressed by reporting more fully the particular Web-crawling policies that have been used. Unfortunately, the challenges to reliability and fuller reporting are compounded when investigators make secondary use (or research using data collected by others; see Stewart & Kamins, 1993) of crawl data originally

Accepted January 23, 2004

 ● Published online 13 August 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20078

collected under unpublished conditions and meant for a different purpose.

The remainder of the article is organized into four main parts. First, some background is given, including a discussion of content crawling versus link crawling that explores some of the procedural distinctions. Second, there is a detailed examination of some particular reliability and validity issues. Third is a report of a Web-crawling experiment that is designed to illustrate the effects of differing crawling policies. Fourth, the conclusion recommends developing a more rigorous approach when reporting investigations using Web-crawl data to facilitate the reproducing of research; there are also some suggestions for further work.

Background and Previous Work

The review by Borgman and Furner (2002) is specifically justified by the growth in electronic communication and the accompanying growth in interest in analyzing its bibliometric properties. The authors identify "the explicit application of bibliometric methods to the study of the World Wide Web" (p. 58). Such methods and their application to informetric analyses of the Web are reviewed by Bar-Ilan (2001) and Bar-Ilan and Peritz (2002). They conclude that existing search engines provide an unreliable basis for informetric analysis of the Web and note that the validity of these informetric findings should be questioned. Mettrop and Nieuwenhuysen (2001) are more emphatic and warn that quantitative analysis of the Web is impaired by the fluctuations and inconsistencies in the responses of search engines which they discovered. Because the difficulties identified could arise from variability in how search engine operators manipulate the results of crawling, they do not necessarily point to a lack of crawling reliability. However, Amitay, Carmel, Darlow, Lempel, and Soffer (2003); Hawking, Craswell, Thistlewaite, and Harman (1999); and Thelwall (2002) all draw attention specifically to the crawler components of search engines. Hawking et al. (1999) observe in respect of their information retrieval (IR) experiment only "that the source of the problem [of poor search engine effectiveness] MAY lie in the spidering" (emphasis added), but Amitay et al. (2003) assert more positively that search engine data "is highly dependent on the configuration [or policy] of the crawler" (p. 5). Thelwall (2002) is concerned with the problem of selecting Web documents for inclusion in crawl-based surveys and argues generally that "different page selection methodologies may yield different results for the same question, and . . . that crawler design parameters can affect survey results" (p. 135). He cautions that information scientists should be aware of the possible differences in sampling by crawling and that small samples can be unrepresentative.

There is little public literature about the details of Web crawling (Cho & Garcia-Molina, 2002; Edwards, McCurley, & Tomlin, 2002). The focus of research by the computer science and information retrieval communities is on content crawling and developing document indexing and retrieval systems. In these circumstances it is desirable to identify duplicated Web documents so that the crawler can ignore them. This is beneficial because it both reduces the computational workload of crawling and improves the indexing and reduces the amount of network traffic and server workload generated by the Web crawler. A prime requirement of Web-crawler design is to conform to Web-crawling ethics (Koster, 1993). These include both respecting privacy requests and minimizing traffic and server workload (see Appendix).

Some Web-Crawling Terminology

The general architecture of a Web crawler is illustrated in Figure 1. The fetcher component requests Web documents (or source nodes) under the direction of the controller. Hyperlinks are extracted from the fetched document by the link extractor and are added to the Web-crawler workload. Hence, the documents (or cited nodes) to which these outlinks refer are themselves fetched.

The summarizer component of the Web-crawler processes the document content and stores information deemed relevant to the purpose of the Web crawler. The summarizer component that undertakes document indexing together with the information store that supports content retrieval are of particular interest to the information retrieval community. This aspect is not discussed further.

If the link-extractor component of the Web crawler discovers cited nodes only as hyperlinks contained in documents composed of hypertext markup language (HTML), then the Web-crawler design is here described as simple. In comparison, an advanced Web-crawler design would extract potential nodes for fetching from additional document formats such as plain text (.txt), Microsoft Word (.doc), and Adobe portable document format (.pdf), as well as from document files that have been compressed. In addition, the crawl space of an advanced crawler design may also extend beyond Web space (or just uniform resource locators [URLs] described by the Internet http and proprietary https schemes) into, for example, ftp or news. Web-crawler research is dominated by the effects on design of the massive scale of the Web. An exhaustive simple crawl of Web space is impractical without making use of theoretically advanced distributed computing techniques (Edwards et al., 2002). Practical research has therefore mainly focused on developing the algorithms needed by the controller component to achieve high-quality crawling using the least computational effort (Cho & Garcia-Molina, 2002). In this context, *high-quality*

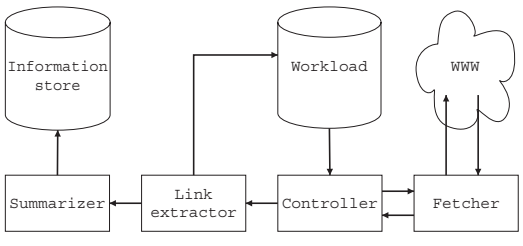

FIG. 1. Web-crawler architecture.

refers to capturing the Web's document content and especially the problem of freshness. This is discussed further under the section on Content Crawling Versus Link Crawling later. Web-crawler design is based on either batch (or periodic) crawling or continuous (or incremental) crawling. A batch Web crawl (or a snapshot of the Web) can be thought of as a single traversal of the Web graph where each node is visited at most once. Continuous Web crawls aim to revisit nodes to recapture and reindex those Web documents that have become stale.

A Web crawl is initialized with a seed set of source nodes to create a workload. In principle, a simple batch crawl then proceeds in the following way:

1. The crawl takes a source node from the workload.
2. If the node has not yet been visited, then the crawl fetches the Web document corresponding to the node.
3. If the Web document is HTML, then the crawl extracts all the cited nodes from the document and adds these to the workload as source nodes.
4. The crawl repeats the procedure from step 1, subject to any design constraint or augmentation until there are no unvisited source nodes left in the workload.

In general, there are three ways to select which source node to take next from the workload. These are called *breadth-first, depth-first,* and *best-first* (e.g., Bergmark, Lagoze, & Sbityakov, 2002). The depth of a cited node is the number of links that the cited node is removed from some relevant source node (Tadi´c, 2002). This source node may be, for example, either a seed node or a home page described by a URL of the form http://some-host.org/. A common crawling design constraint is to set a depth limit to the crawl. Figure 2 illustrates a small hierarchy of connected documents.

Given a crawl starting with document *a*, then a breadth-first crawl prioritizes fetching of the six nodes in the sequence, *a*, *b*, *c*, *d*, *e*, and *f*. In comparison, a depth-first crawl would aim to fetch the six nodes in the sequence *a*, *b*, *d*, *f*, *e*, and then *c*. Under the constraint of a depth limit of two, then document *f* would not be included in either a breadth- or depth-first crawl. Breadth-first is an attractive crawling policy because it is computationally simple to implement and, compared with depth-first, is more likely to avoid overloading individual servers (Chakrabarti, Joshi, Punera, & Pennock, 2002).

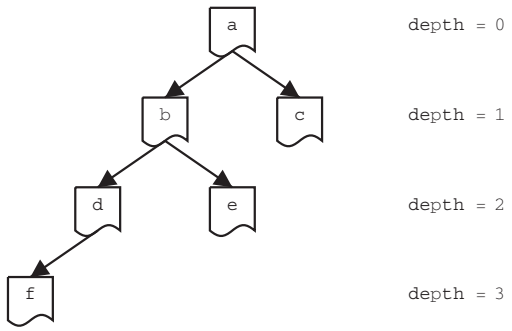

FIG. 2. Document depth.

Best-first-fetching policies such as making use of "PageRank" (Brin & Page, 1998) to select the next document to be fetched aim to reduce the overall computational and network workload when content crawling by predicting those Web pages that are likely to be of high quality. Najork and Wiener (2001), however, found that in respect of content quality, more sophisticated best-first-fetching policies such as PageRank do not, in practice, offer an advantage over breadth-first. They speculate (p. 117) that this is because "the most important pages have many links to them from numerous hosts, and those links will be found early, regardless of on which host or page the crawl originates."

If, as may commonly be the case (Cho & Garcia-Molina, 2002), there is an overall limit on the total number of nodes traversed during a crawl or a restriction on the duration of a crawl, then breadth-first-, depth-first-, and best-first-based prioritizing of node fetching will yield different results. In the preceding example, if an overall limit of five is imposed, then the breadth-first crawl would exclude document *e*, compared with excluding document *c* for a depth-first crawl.

Designers of content crawlers are interested in identifying both exactly duplicated Web pages and also those pages that are near duplicates (Henzinger, 2003). Duplicated Web pages can occur when the content file of the document is explicitly copied and stored in more than one location (and possibly on more than one host server), and when different URLs are used as citations to link to the same content file. For example, it is often found that the two citations of the form http://foo.com/ and http://foo.com/index.html link to the same file and so have identical content. A special class of duplicate or near duplicate Web pages are so-called mirrored hosts. Bharat, Broder, Dean, and Henzinger (2000) evaluate techniques to detect and eliminate these duplicates to gain the benefits of avoiding storing the duplicated content and of perturbing and degrading the Web-graph model.

The motivation for the final document-based crawling-design constraint mentioned here arises because some of the nodes that a crawler encounters anticipate a real user submitting data to a server. The URLs of such nodes commonly include a "?," "cgi-bin," or "&." Because these nodes have the potential to disrupt the operation of a Web crawler, the crawling policy may be to ignore them (Chakrabarti et al., 2002).

A variation to crawling the entire Web space according to, say, a breadth-first policy, is to partition or divide the graph of the crawl space into subgraphs and to crawl each subgraph independently. A convenient partitioning method is, for example, to include or exclude nodes depending on the host domain name in their URLs. Hence, all the URLs having a particular host domain name constitute a partition. The crawl proceeds partition by partition, although within each partition the Web crawling may still be breadth-first. The Internet Archive crawler is designed to operate by partition (i.e., site-by-site) and has no bias toward high-quality pages (Burner, 1997; Najork & Wiener, 2001). It is also reported by Burner that this approach is well suited to support the Archive's goal of exhaustive coverage. An obvious question

is what happens in respect of the links discovered that are to URLs outside the partition. In the case of the Archive, these interpartition links are stored and used to identify new sites or hosts for crawling.

A Web-crawling policy can also augment rather than constrain the fetcher. For example, in Figure 2 the URL of the cited node *b* linked to from the source node *a* may be of the form http://different-host.org/one/two/page.html. A simple Web crawler just extracts this URL and adds it to the workload. An augmented simple Web crawler might also add one or more of the three URLs http://different-host.org/one/two/, http://different-host.org/one/, and http://different-host.org/ that can be derived from the citation. A Web-crawling policy that uses all the possible derived nodes is described here as being augmented by path ascent. Such a path-ascending Web crawler has an opportunity to discover otherwise unconnected nodes as described later in the section, Web-Crawling Reliability and Validity.

Content Crawling Versus Link Crawling

There appear to be two distinctions that can be drawn between these different classes of Web crawlers, which may be used to characterize them. First, Web crawlers designed to discover Web content, that is, content crawlers, include features to avoid duplicate and near duplicate Web documents. In contrast, link crawlers would not avoid Web documents just because their content is duplicated (although they may be avoided for other reasons). From a search engine provider's point of view, not collecting duplicate Web pages is advantageous because it reduces redundant document summarization and storage, and because the version of the Web-link graph that is used to refine the content-retrieval procedures works better when the links from duplicated documents are eliminated.

Second, Web crawlers designed to collect content also appear to have a goal of maintaining content freshness. Some genres of Web document such as those involving news or broadcasting schedules rapidly become stale as their content becomes out of date. Clearly Web pages that have rapidly changing content will be more prone to becoming stale. Freshness can thus be thought of as a measure of the extent to which a search engine has fetched the most recently changed version of Web documents.

Hence, content crawling appears to be undertaken continuously, whereas the periodic (but extended and exhaustive) batch crawls undertaken by the Internet Archive exemplify link crawling. Continuous Web crawling leads to a potential problem with how obsolete or stale documents are managed. The search engine providers' goal is to offer relevant, fresh Web pages in response to a user's query. But it is not clear what happens to stale document versions, which may contain obsolete or duplicated citations, when documents in search engine IR databases are refreshed. On the one hand, these may be removed, but on the other, they may just be less preferred (when determining relevance). In consequence, overall document (and link) counts may be double-counting refreshed documents by including also the links in obsolete and stale documents.

It has been widely reported that search engines' coverage of the Web is modest and that there is little overlap between them (Bharat & Broder, 1998; Lawrence & Giles, 1998, 1999). At the same time "all [search] engines have a bias towards well-connected, content-rich pages" (Bharat & Broder, 1998, p. 388). The inference from this, that different content crawlers generate differently biased samples of the Web, is supported by Henzinger (2001, p. 46), who is in a position to state that "the [content] crawlers of different search engines make different decisions, and so collect different sets of Web documents. For example, a crawler might try to preferentially crawl 'high quality' Webpages."

The Web-crawling algorithms or policies used to define *high quality* and to control the selection of content-crawl samples of Web documents are not published (Cho & Garcia-Molina, 2002; Najork & Wiener, 2001). Therefore, users carrying out secondary research are not informed as to the selection basis and bias of the sample.

In principle, this selectivity particularly affects informetric link analysis of the Web because the version of the Web graph developed as an adjunct to content crawling is likely not to copy the real Web or even be a subset of it. It is instead a proprietary interpretation of the real Web graph. Whether this distinction between Web graphs is material will depend on the exact nature of informetric research being undertaken.

Web-Crawling Reliability and Validity

Some specific Web-crawling reliability and validity issues that arise generally are now discussed in respect of the Web graph represented in Figure 3. This shows a crawl space of two partitions, A and B, which could represent different hosts or domains. Web-crawling inconsistencies will occur depending on the crawling policies adopted.

In this example, it is assumed that the 19 documents shown are all different except the pair marked *m*, which mirror or duplicate each other. A simple unconstrained content crawl over the entire crawl space would, at best, traverse 17 of these documents. This is because the isolated (or island)

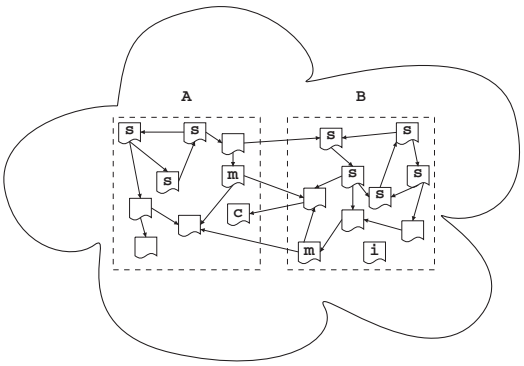

FIG. 3. Example of crawl space.

document marked *i* would not be discovered, and only one, the first to be discovered, of the duplicated pages would be fetched. Which of the two documents this would be would depend on external influences such as server response times, so that repeated identical content crawls could show either two links from partition A to B and one from B to A or one link from A to B and two from B to A. Hence, a link analysis based on this content crawl is not reliable; a pair of researchers employing an identical technique obtain different results. In this example it is also clear that content-crawl-based link data does not validly represent the interpartition link structure of the crawl space even though it may validly represent the crawl space's content.

An equivalent simple unconstrained link crawl would discover 18 documents and validly give the interpartition link structure as two links from A to B and two links from B to A. The isolated document would again prove elusive.

Hence, a third researcher who uses link-crawl-based data in respect of the same crawl space arrives at a third and different answer to the question of how many links are there between A and B.

It may appear that document *i* cannot be discovered because it has no inlinks. However, depending on the server path structure of the documents and whether or not server directory pages are revealed, a path-ascending crawler may find *i* because it would be included as an outlink in its parent directory. Discovering isolated documents could be especially important to a link analysis if they contain outlinks, even if these were only to other documents within the same partition.

In the event that a crawl seed set in respect of the example crawl space fails to include any of the documents marked *s*, then the coverage obtained by either a simple content crawl or simple link crawl over the crawl space would reduce. In the worst case, no additional document would be discovered. At best, the simple link crawl would find 14 additional documents compared to 13 for an identically seeded simple content crawler. This wide variation in potential node discovery suggests that crawls on real Web graphs may be sensitive to the size of the seed set as well as crawl policy. An investigation of variation according to seed set size is included as part of the Web-crawling experiment reported in the next section.

The discussion of Web-crawling reliability and validity has so far assumed a crawl of the entire example crawl space shown in Figure 3. Content crawling produces two different answers, while link crawling produces a third answer. Partition-based crawling of this crawl space could introduce a further loss of reliability depending on how the interpartition link information is processed. Document *c* is conditionally isolated, that is, it is cited only by a document in a different partition, B, and will not be discovered by a simple crawl of partition A; *c* will only be discovered by a simple crawl if the interpartition outlink to it from partition B is fully exploited. (Using the outlink to identify, say, just the host serving partition A is not sufficient.) Hence, a partitioned simple crawl of the crawl space may produce a different analysis for partition A compared to a nonpartitioned simple crawl of the entire crawl space. The former may exclude the conditionally isolated document *c*, whereas the latter does not.

The example shows that the differences in principle between the two classes of Web crawler, content crawler and link crawler, have a practical effect and results in different answers when applied to the problem of analyzing the link structure of the example crawl space. That is, the broad technique of Web crawling is not reliable in respect of determining link structure. The corollary is that a difference that is found between two apparently equivalent investigations of link structure could reflect the data collection technique's lack of reliability and not a real difference.

The Experiment

Objective

The Web-crawling experiment undertaken is based on using just a single class of crawler (content) rather than comparing the two classes of crawler as in the previous section. The experiment is designed to demonstrate the difference in effect, if any, of a variety of crawling policies or operational characteristics. The objective is to discover whether or not differences occur, not to evaluate or recommend any policy. Hence, any comparative advantage found for any policy may be particular to the subgraph of the Web used in the experiment.

Seven simple batch Web-crawling policies are investigated. These are

1. Augment by path ascent,
2. No constraints,
3. Constrain to the default server port for source nodes,
4. Constrain to exclude "query" URLs for source nodes,
5. Constrain to a maximum depth of 9,
6. Constrain to a maximum depth of 6, and
7. Constrain to a maximum depth of 3.

The no-constraint policy is essentially a control or default crawler. This is compared with three identical crawlers but for one changed characteristic. These affect the server port (which was thought possibly relevant because the experimental subgraph lies inside a protective firewall), whether or not the URL has a "query," and the depth constraint. These two characteristics are chosen because they occur as examples of constraints used in practice. The values selected for the depth constraint are chosen to provide a good range of values.

The augmented crawler that uses path ascent to aid node discovery is a novel approach to crawling when compared with the usual description of how Web crawlers operate. It attempts, in simulation, to read the server's directory structure to more rapidly discover nodes and to discover nodes that otherwise have no inlinks.

A subsidiary objective of the experiment is to investigate how crawls respond to a variation in the size of the initial

TABLE 1. Characteristics of the reference subgraph.

Source nodes	Source node frequency	Cited node frequency: To within <wlv.ac.uk>	Cited node frequency: To elsewhere	Cited node frequency: Total
All	123,375	302,680	36,277	338,957
Not having a query URL	122,479	301,353	36,253	337,606
On default port	121,466	299,946	35,894	335,840

seed set. In general, it is reported that the bigger the seed set the better. However, additional seeds that are themselves connected just to existing seeds do not improve the efficacy of the overall seed set.

The experiment makes use of Web-crawl simulations. Although individual simulations have been used in the context of investigation computing performance, comparative investigations of the effects of different crawling policies have not been reported, nor have Monte-Carlo approaches been applied to the study of Web crawling.

Method

The experimental design uses a Monte-Carlo approach and simulated Web crawls over a fixed reference subgraph. The Web-crawl simulations are conducted using crawl seed sets of randomly selected seed source nodes. The size n of the seed sets varies from $n = 50$ to $n = 1{,}000$. This is to reveal how sensitive crawls may be to the number of initializing source nodes. Thirty simulated Web crawls were carried out over the reference subgraph of the Web graph for 20 values of n and for each of the seven Web-crawling policies investigated. Hence, in total 42,000 simulated Web crawls have been processed.

A preliminary reference subgraph was produced by a simple batch content crawl of the wlv.ac.uk subdomain of Web space using a purpose configured version of the publicly available Harvest-NG Web crawler (Harvest-NG, 2003). The preliminary crawl data were then processed to generate URLs corresponding to each directory. For example, the URL http://wlv.ac.uk/a/b/c/file.html would generate the three URLs http://wlv.ac.uk/a/, http://wlv.ac.uk/a/b/, and http://wlv.ac.uk/a/b/c/. These, together with the preliminary subgraph source nodes from the subdomain, formed the final seed set of about 39,000 nodes, and the subdomain was content-crawled a second time to produce the reference subgraph.

Detailed crawl configuration and procedural information is given in the Appendix. In particular, source nodes given by "query" URLs, that is, URLs that contain a "?", are not generally excluded.

It should be noted that because the reference subgraph has been augmented, then it contains a node corresponding to every directory listing that is accessible with a generated URL, regardless of whether the directory listing has an inlink from elsewhere in the subgraph. In general, these directory lists are therefore isolated.

The reference subgraph consists of about 123,000 source nodes that contain outlinks to about 339,000 cited nodes. Because the reference link crawl was undertaken inside a protective firewall, several Web servers run on ports other than the default port (i.e., port 80 for http). Not all server ports are accessible from outside the firewall.

Table 1 summarizes the main characteristics of the reference subgraph.

The number of source nodes (about 123,000) given in Table 1 is inflated by the augmented crawling procedure that generated the reference subgraph. This is because directory citations of the form http://*wlv.ac.uk/*/, where the server reveals a file directory are included even when there is no explicit citation to this node within the Web graph.

The number of cited nodes within the subdomain (about 303,000) is about 90% of the nodes cited. The numerical difference between the frequency of source nodes and cited nodes within the subdomain reflects the simple crawling procedure in which source nodes must be in html format. Although image files compose the majority of the difference, there are also non-HTML Web document formats, for example, Microsoft Word, which could have potential as a source of outlinks.

The overall ratio of cited nodes to source nodes is about 2.75. That is, on average, each source Web document has 2.75 outlinks. However, both this proportion and the 90% proportion above are inflated by the way in which the reference subgraph has been constructed. Work by Najork and Heydon (2001, p. 7) suggests that about 80% of outlinks are to other nodes within the same partition.

Results

To compare the result of each simulated Web crawl, the total number (frequency) of source nodes and cited nodes discovered by the crawl is counted. The arithmetic means of the 30 such frequencies for each crawl seed set size, n, are plotted as a function of n in the graphs shown in Figures 4 and 5. The functions in respect of each crawling policy investigated are separately identified. In addition, Table 2 reports for each of the seven Web-crawling policies investigated the mean and standard error of source and cited nodes discovered and the ratio of cited nodes per source node when $n = 1{,}000$.

The overall appearance of the source node discovery functions illustrates how for all the simple crawling policies the functions grow, but possibly asymptotically, to some

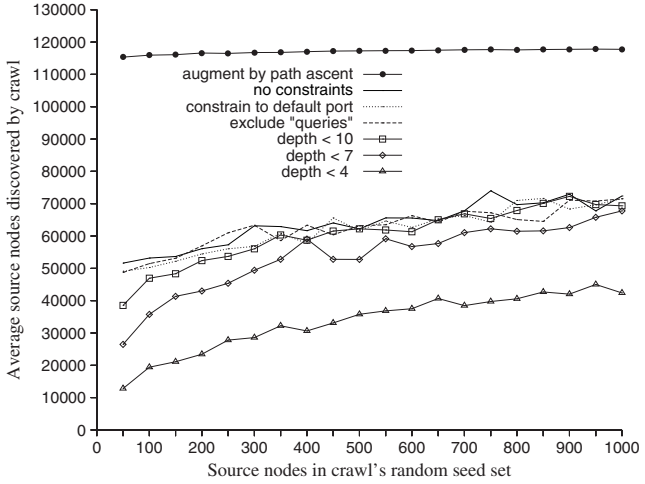

FIG. 4. Source node discovery by crawling policy.

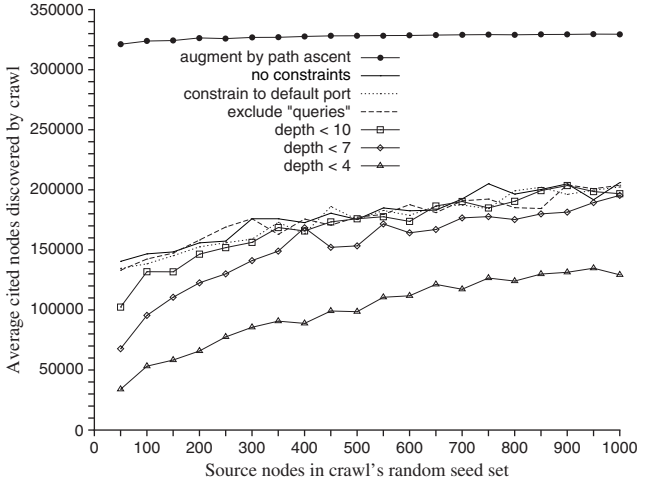

FIG. 5. Cited node discovery by crawling policy.

limiting value. This shows a diminishing return for source node discovery in relation to the computational and network resource that must be expended as the crawl progresses. The goal of search engine crawlers is to obtain all their high-quality Web documents early in their content crawl so that their crawl can then be truncated.

The exceptional source node discovery function is in respect of the augmented crawling policy. This crawler is not as sensitive to the size of the seed set as the other crawl policies. This is because, given that the crawl is able to fetch a directory list, all the files or documents in the directory can be identified immediately by the crawler regardless of how good (or bad) is the linkage among them.

The big difference in the value of the function is caused by the large number of isolated documents in the reference subgraph. This is an artifact caused by the way in which the subgraph was constructed in that most of the nodes corresponding to the directory lists are isolated. If this were not the case, then the control unconstrained crawler would have been expected to behave like the augmented crawler because it too would have had access to directory lists.

The cited node discovery functions correspond approximately (but not exactly) to their underlying source node function. Hence, for the reference subgraph, although there is the same diminishing return, the source nodes that are more computationally strenuous to discover do continue to provide additional outlinks or citations. The alternative could have been that source nodes all draw their citations from a limited population, which is soon completely revealed. The cited node discovery function for the augmented crawling policy is necessarily extreme and always reports almost the entire subgraph as described in Table 2.

Table 2 shows that at $n = 1{,}000$, the control crawl, crawling without constraint, discovers 72,390 source nodes. The comparative performance of the other constrained crawls reveals differences. In particular, the most depth-constrained crawl clearly performs differently in discovering only 42,391 source nodes. The difference between these two crawls is statistically significant for all the performance measures shown, source nodes, cited nodes, and cited nodes per source node.

The reduction in the ratio of cited node per source node that is observed as the depth constraint is relaxed corresponds to a thinning out of citations as the subgraph is more completely crawled. That is, it suggests that documents topologically nearer the home page are numerically richer in outlinks. This is part of the rationale underpinning an explanation of the strengths of breadth-first content crawling mentioned earlier (Najork & Wiener, 2001).

Hence, the results of the Web-crawling experiment as illustrated in Figures 4 and 5 demonstrate that Web crawling according to different crawl policies produces different

TABLE 2. Effect of crawling policy.

Crawling policy	Source nodes (at $n = 1{,}000$)	Cited nodes (at $n = 1{,}000$)	Cited nodes per source node
Augmented by path ascent	117,770 (se = 51)	329,561 (se = 110)	2.80 (se = 0.0006)
No constraints	72,390 (se = 1,879)	205,974 (se = 4,260)	2.85 (se = 0.02)
Default port source only	71,564 (se = 2,320)	202,325 (se = 5,143)	2.84 (se = 0.03)
Exclude "query" sources	71,481 (se = 2,050)	203,837 (se = 4,837)	2.86 (se = 0.02)
Crawl depth < 10	69,346 (se = 2,139)	196,770 (se = 5,163)	2.85 (se = 0.02)
Crawl depth < 7	67,786 (se = 2,651)	195,377 (se = 6,209)	2.90 (se = 0.03)
Crawl depth < 4	42,391 (se = 663)	129,196 (se = 3,024)	3.04 (se = 0.04)

outcomes. Although this difference is moderated to some extent by the size of the seed set, there remains a significant difference between the control crawl and the most depth-constrained crawl. In the absence of information about the crawl policy being revealed, then two researchers using data collected with different policies would erroneously conclude that, for example, a difference in cited nodes per source node had been discovered.

The subsidiary objective of the experiment is to examine how crawls vary according to the size of the seed set. Within the range considered, the results show that the outcome of simple Web crawling is generally sensitive to the size of seed set. For the reasons discussed previously, this is not the case for the augmented crawl.

The ultimate similarity in outcome of five of the seven crawl policies investigated suggests that having a large seed set, or more precisely, a seed set that is a good representation of sources in the crawl space, compensates for some of the crawl constraints considered. However, there appears to be a critical depth value within which the results of a simple crawl is comparatively limited. For the wlv.ac.uk subdomain, this value is less than six. When the maximum depth of the crawl is six, the ultimate result (at $n = 1{,}000$) approximates to the unconstrained crawl, but, for example, when the maximum depth is three, the ultimate result is materially different. The lack of an ultimate difference between the crawls constrained to the default server port or to exclude queries also reflects a particular characteristic of the wlv.ac.uk subdomain. Table 1 shows that about 99% of source nodes fall into these categories; therefore, an ultimate difference in the result of a crawl becomes progressively infeasible as the crawl coverage increases.

The source nodes in the seed sets used here were chosen randomly. In practice, the source nodes in Web-crawling seed sets can be carefully selected. This suggests that a possible mechanism to explain both Najork and Wiener's finding and the lack of overlap between different search engines is that their Web crawling was within a critical depth where crawl outcome is sensitive to the choice of seed set, especially when this is small. Najork and Wiener (2001) themselves suggest that most high-quality Web documents (i.e., the target of content crawling) are at small depths. The potential for inconsistencies in crawl results that can arise in these circumstances is illustrated by the 30 experimental simulations when the depth limit was three and the seed set size $n = 50$. Although the average number of cited nodes discovered was 33,889, this crawl result ranged from 17,084 to 62,509. The wide variation here provides the potential for a lack of overlap as is observed with actual search engines.

The result functions of the augmented Web-crawling policy differ both in respect of a lack of seed set size sensitivity and in magnitude. An important effect of augmentation by path ascent is that the seed set is implicitly expanded to include the home page of each Web server represented. This maximizes the opportunity of the Web crawler to exploit any natural hierarchy in the URL structure of Web documents. Hence, every large seed set effectively generates the same home page down crawl of the entire crawl space. This top-down crawl includes discovering isolated documents from server directory information whenever this is possible. The number of such documents is inflated because many of the directory pages will have no links to them but have been obtained only by the augmentation procedure.

A second contribution to the difference is that documents in the subdomain cited only as, say, http://host.wlv.ac.uk/path/index.html, are included additionally in the reference subgraph as http://host.wlv.ac.uk/path/, which is isolated and therefore not discoverable by a simple crawl (see the Appendix).

Discussion

The experiment just described is a novel investigation of Web-crawler behavior both in that multiple (42,000 in this case) systematic crawls of the same graph have not been previously reported nor has the effect of crawl policies on the resulting Web graph been compared. The earlier analysis of the example crawl space showed that the different classes of Web crawler, that is, content crawlers and link crawlers, are inconsistent and that content crawlers do not reliably provide link structure data. The experiment focuses on just one class of crawler and shows in particular that the Web crawlers investigated are inconsistent in their results. They are therefore not reliable. In particular, crawlers that are identical except for a difference in their depth policy produce different results. Therefore, in respect of the experiment's objective, it is shown that a crawler's policy does matter; changing the policy can change the result. Hence, to interpret or compare analyses of crawl-based data, one must know the crawl policy under which the data was selected.

That is, Web crawling as a description of a data selection technique is inadequate in that it encompasses too many hidden variables (the class, policy, seed, etc.) and in the absence of additional qualification leads to inconsistent results. Thus, Web crawling is, of itself, not reliable.

However, the experiment also shows that, for the subdomain concerned and given a sufficient seed set, any one crawl policy from those investigated converges and produces consistent results. The two most consistent policies are the augmented crawler and the most depth-constrained crawler (source node standard error = 51 and 663, respectively). This shows that reliable Web crawling is possible and is achieved by more fully disclosing or reporting the hidden variables.

Two questions concerning generalizability arise. First, is Web crawling, of itself, generally not reliable? Second, to what extent might fully reported Web crawling be generally reliable?

Although the reference subgraph is derived from only a single academic institution in the United Kingdom, the argument developed here is that by both critical examination and by direct experiment using a real subgraph of the Web, examples of inconsistency can be demonstrated. Because any graph that contains the subgraph will also produce the

inconsistency found, even if this is confined to the subgraph, then the lack of reliability remains and is general.

For the subgraph investigated in the experiment, all the simulated crawlers taken individually and with a sufficient seed set produced consistent results. That is, when fully reported, Web crawls of the subgraph are reliable. Moreover, it can be conjectured that some policies are equivalent, for example, unconstrained and large-depth (>10) limited crawling. Although it is highly improbable that other subgraphs would not produce similar results when using the same crawler, the real difficulty is in providing a complete definition of all the characteristics of a particular crawler so that the same selection technique (crawler) can be used by another researcher. Otherwise, research results remain crawler-specific.

Conclusion

In this article, I set out to investigate the reliability of Web crawling and particularly Web crawling in the context of the use of crawl data to support informetric studies of the Web such as link structure analyses. Two classes of crawler, content and link, have been identified. The investigation has identified by both critical examination and experiment that samples of the Web obtained by Web crawling will be biased according to the class of crawler and the value of several Web-crawling characteristics. Collectively these characteristics (the crawl policy) can be used to describe and define the particular crawler with which any given data sample was selected. In the absence of the full reporting of the crawl policy, data samples cannot be reliably reproduced. However, reliable Web crawling has been demonstrated when the policy of the crawler is known.

Historically, crawling policies, including those of the major search engines, have not been reported. Secondary users of crawl-based data, such as informetric researchers studying the Web, are especially disadvantaged because they are not informed as to how the data in their datasets have been selected, if the selection basis has changed over time, or how the selection for one source compares with that for another source.

In consequence, the affected aspects of informetric research into the Web cannot sustain critical examination or be reproduced, whether this be to support or to refute any finding. Unfortunately, it is therefore also unclear whether or not any of the bias and selectivity in sampling by the major search engines actually do modify informetric findings (see the section on Future Work). Because the potential effect must depend on the detail of each investigation's research question and methodology, then the crawling policies by which relevant samples are obtained should be reported. This will improve the reliability of informetric analyses of the Web graph and in particular enable reproduction of investigative methodologies. The secondary use of crawl data in the absence of this fuller reporting also introduces the possibility that separate studies will unwittingly be based on the same data such as is suggested by Bailey et al. (2003, p. 9).

Future Work

The paucity of detailed design and operational performance information about Web crawlers means that there is little evidence for determining how material might be the effect of a particular crawling policy. In general, therefore, future work in this area should aim at determining how sensitive Web crawling is to changes in the crawler's characteristics.

The experiment undertaken indicated the existence of a critical crawl depth value occurring in combination with a random seed set within a particular subgraph. This suggests that other subgraphs should be investigated as well as considering critical crawl depths with other types of seed sets, such as home pages. Depth-constrained simple link crawls from seed sets comprising a list of home pages are reproducible and could therefore provide a basis for reliable comparison.

The issue of duplicate avoidance by content crawlers has been discussed here, but as yet there has been no investigation into how this procedure affects the resulting Web graph. Given the central role that duplicate avoidance appears to have as regards content crawling and the potential differences that uncontrolled duplicate avoidance could have on informetric analyses of link structure, then such investigations should receive priority.

Acknowledgments

This work was supported by a grant from the Common Basis for Science, Technology and Innovation Indicators part of the Improving Human Research Potential specific programmed of the Fifth Framework for Research and Technology development of the European Commission. It is part of the Web indicators for scientific, technological, and innovation research (WISER) project, (Contract HPV2-CT-2002–00015).

References

Amitay, E., Carmel, D., Darlow, A., Lempel, R., & Soffer, A. (2003). The connectivity sonar: Detecting site functionality by structural patterns. Proceedings of the 14th Conference on Hypertext and Hypermedia (pp. 38–47), Nottingham, UK, August 26–30, 2003. New York: ACM.

Bailey, P., Craswell, N., & Hawking, D. (2003). Engineering a multi-purpose test collection for Web retrieval experiments. Information Processing and Management, 39(6), 853–871.

Bar-Ilan, J. (2001). Data collection methods on the Web for informetric purposes: A review and analysis. Scientometrics, 50(1), 7–32.

Bar-Ilan, J., & Peritz, B.C. (2002). Informetric theories and methods for exploring the Internet: An analytical survey of recent research literature. Library Trends, 50(3), 371–392.

Bergmark, D., Lagoze, C., & Sbityakov, A. (2002). Focused crawls, tunneling, and digital libraries. In V. Agosti & C. Thanos (Eds.), Proceedings of the Sixth European Conference on Digital Libraries (pp. 91–106), Rome, Italy, September 16–18, 2002. London: Springer.

Bharat, K., & Broder, A. (1998). A technique for measuring the relative size and overlap of public Web search engines. Computer Networks and ISDN Systems, 30(1–7), 379–388.

Bharat, K., Broder, A., Dean, J., & Henzinger, M.R. (2000). A comparison of techniques to find mirrored hosts on the WWW. Journal of the

American Society for Information Science and Technology, 51(12), 1114–1122.
Borgman, C.L., & Furner, J. (2002). Scholarly communication and bibliometrics. In B. Cronin (Ed.), Annual review of information science and technology: 36, (chap. 1, pp. 3–94). Medford, NJ: Information Today for American Society for Information Science & Technology.
Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1–7), 107–117.
Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the Web. Computer Networks, 33(1–6), 309–320.
Burner, M. (1997). Crawling towards eternity: Building an archive of the World Wide Web. New Architect: Internet Strategies for Technology Leaders, 2(5). Retrieved May 17, 2003, from http://www.newarchitectmag.com/archives/1997/05/burner/
Chakrabarti, S., Joshi, M.M., Punera, K., & Pennock, D.M. (2002). The structure of broad topics on the Web. Proceedings of the 11th World Wide Web Conference (pp. 508–516), Honolulu, Hawaii, May 7–11, 2002. New York: ACM.
Cho, J., & Garcia-Molina, H. (2002). Parallel crawlers. Proceedings of the 11th World Wide Web Conference (pp. 124–134), Honolulu, Hawaii, May 7–11, 2002. New York: ACM.
Cronin, B. (2001). Bibliometrics and beyond: Some thoughts on Web-based citation analysis. Journal of Information Science, 27(1), 1–7.
Edwards, J., McCurley, K., & Tomlin, J. (2002). An adaptive model for optimizing performance of an incremental Web crawler. Proceedings of the 11th World Wide Web Conference (pp. 106–113), Honolulu, Hawaii, May 7–11, 2002. New York: ACM.
Egghe, L., & Rousseau, R. (1990). Introduction to informetrics: Quantitative methods in library, documentation and information science. Oxford: Elsevier.
Eichmann, D. (1994). The RBSE spider: Balancing effective search against Web load. In Proceedings of the First World Wide Web Conference, Geneva, Switzerland, May 25–27, 1994. Retrieved May 12, 2003, from http://mingo.info-science.uiowa.edu/eichmann/www94/Spider/Spider.html
Gordon, M., & Pathak, P. (1999). Finding information on the World Wide Web: The retrieval effectiveness of search engines. Information Processing and Management, 35(2), 141–180.
Harvest-NG. (2003). Harvest-NG homepage. Retrieved June 18, 2003, from http://webharvest.sourceforge.net/ng/
Hawking, D., Craswell, N., Thistlewaite, P., & Harman, D. (1999). Results and challenges in Websearch evaluation. Computer Networks, 31(11–16), 1321–1330.
Henzinger, M.R. (2001). Hyperlink analysis for the Web. IEEE Internet Computing, 5(1), 45–50.
Henzinger, M.R. (2003). Algorithmic challenges in Web search engines. Internet Mathematics, 1(1), 115–126. Retrieved September 10, 2003, from http://www.internetmathematics.org/volumes/1/1/pp115_123.pdf
Kerlinger, F.N., & Lee, H.B. (2000). Foundations of behavioral research (4th ed.). London: Harcourt College.
Kleinberg, J.M., Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A.S. (1999). The Web as a graph: measurements, models, and methods. In T. Asano, H. Imai, D.T. Lee, S.-I. Nakano, & T. Tokuyama (Eds.). Proceedings of the 5th International Conference on Computing and Combinatorics (pp. 1–17), Tokyo, Japan, July 26–28, 1999. London: Springer.
Koster, M. (1993). Guidelines for robot writers. Retrieved May 12, 2003, from http://www.robotstxt.org/wc/guidelines.html
Koster, M. (2003). The Web robots page. Retrieved May 12, 2003, from http://www.robotstxt.org/wc/robots.html
Lawrence, S., & Giles, C.L. (1998). Searching the World Wide Web. Science, 280, 98–100.
Lawrence, S., & Giles, C.L. (1999). Accessibility of information on the Web. Nature, 400, 107–109.
Meghabghab, G. (2001). Google's Web page ranking applied to different topological Web graph structures. Journal of the American Society for Information Science and Technology, 52(9), 736–747.
Mettrop, W., & Nieuwenhuysen, P. (2001). Internet search engines: Fluctuations in document accessibility. Journal of Documentation, 57(5), 623–651.
Najork, M., & Heydon, A. (2001). High-performance Web crawling. Research Report 173, Compaq: Systems Research Center, Palo Alto, California. Retrieved December 10, 2003, from ftp://gatekeeper.research.compaq.com/pub/DEC/SRC/research-reports/SRC-173.pdf
Najork, M., & Wiener, J.L. (2001). Breadth-first crawling yields high-quality pages. Proceedings of the 10th World Wide Web Conference (pp. 114–118), Hong Kong, May 1–5, 2001. New York: ACM.
Stewart, D.W., & Kamins, M.A. (1993). Secondary research: Information sources and methods (2nd ed.). London: Sage.
Tadić, B. (2002). Growth and structure of the World Wide Web: Towards realistic modelling. Computer Physics Communications, 147(1–2), 586–589.
Thelwall, M. (2001). A Web crawler design for data mining. Journal of Information Science, 27(5), 319–325.
Thelwall, M. (2002). Methodologies for crawler based surveys. Internet Research: Electronic Net-Working Applications and Policy, 12(2), 124–138.
Wall, L., Christiansen, T., & Schwartz, R.L. (1996). Programming Perl (2nd ed.). Cambridge, MA: O'Reilly.

Appendix

The appendix provides detailed crawl configuration information and describes the procedure that was followed to construct the reference subgraph of the wlv.ac.uk subdomain of the World Wide Web.

The Web-crawling program used was Harvest-NG (Harvest-NG, 2003), which is designed to be highly configurable. Harvest-NG consists of several Perl modules (Wall, Christiansen, & Schwartz, 1996), which additionally facilitates customizing the crawler to meet the specific requirements of an investigation. The crawler was configured as a simple batch content crawler of the Web space within the University of Wolverhampton subdomain of the UK academic network:

1. Links to be added to the workload were extracted just from files comprising HTML.
2. The crawl was implemented so that each document was at most fetched once. To ensure this, the crawler maintained a record of the URL of each node in a unique standard form so that, for example, the URL http://HOST/File was equivalent to http://host/File.
3. Files containing duplicated content were avoided.
4. Only source nodes having URLs with http or https schemes were fetched.
5. Only source nodes having URLs where the host part is a host within the wlv.ac.uk subdomain were fetched. (A document node having a URL within the subdomain in dotted decimal format such as http://134.220.1.46/ was therefore excluded.)

The crawler complied with the informal robots exclusion protocol, and it observed the meta tag robot privacy protocol (Koster, 2003). Ethical Web crawling also requires that files that are not processed should not be fetched. This reduces unnecessary network and server workload. Therefore, none of the files with file extensions corresponding to any of the

following were fetched: png, gif, jpg, jpeg, tiff, bmp, svg, css, ra, doc, rtf, ppt, xls, exe, cnf, lck, btr, sav, tex, pl, py, pm, c, h, ps, pdf, zip, gz, gzip, Z, tar, and tgz.

The crawler described itself as just "Mozilla/5.0" to emulate the Mozilla browser. Some Web servers may adjust their response, or in the extreme fail to respond, according to the browser from which the request originates.

Source nodes with URL path components including cgi-bin or www-bin were not fetched. This is to avoid disrupting the crawler, which might otherwise endlessly (or recursively) attempt to fetch documents from a Web-accessible database. However, documents otherwise identified by query URLs (or URLs that contain a "?") were fetched.

In general, there is a recursion problem when Web crawling if by accident or design a particular Web document is constructed so as to generate an infinite sequence of cited URLs to be fetched. An example of the deliberate construction of such a "spider trap" found elsewhere included the comment "Chew on this AltaVista" (J.S. Katz, personal communication, July 2003). The crawl to construct the reference subgraph was protected from recursion by not fetching any source node where a URL path component is repeated more than four times.

The crawling procedure was to first carry out a preliminary crawl to generate a path-ascending augmented seed set for the reference subgraph crawl. The preliminary crawl used the home page of a major Web server within the subdomain as its seed because such a crawl was known to reach all the Web servers within the subdomain (M. Thelwall, personal communication, March 2003). All the source and cited nodes discovered by the preliminary crawl were then analyzed and augmented by path ascent. The resulting 39,000 nodes were used to seed the crawler that was used to produce the investigation's subgraph. All the directory citations or URLs having the form http://*wlv.ac.uk/*/ that were expressly created by the augmentation rather than by being discovered as a citation thus represent isolated documents within the subgraph because no node has outlinks to them.

The crawler attempted to fetch directory lists in a variety of formats that are offered by the Web server. These additional redundant source nodes, such as http://*wlv.ac.uk/*/?N=D, were excluded.

The source nodes and all the cited nodes that they contain provided the reference subgraph. This was stored as a fixed database and was used by the crawl simulations so that the same reference subgraph is used throughout the experiment.