Characterizing and Mining Citation Graph of Computer Science Literature

by

Yuan An

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2001

Abstract

Computer science literature, as many other natural systems behave, form a directed graph–we call it Citation Graph of Computer Science Literature, whose nodes are articles and edges are links to the articles cited in a paper. With hundreds and thousands of publications getting published each year in computer science, people are more interested in exploring the features hidden behind such huge directed graph by modern graph-theoretic techniques. In this study, we constructed a web robot querying the prominent computer science digital library *ResearchIndex* to build citation graphs. With the reasonable size citation graph in hand, we first verified that the in-degrees of nodes(i.e., the citations of articles) follow the Power law distribution. Next, we apply a series graph theoretic algorithms on it: *Weakly Connected Component*, *Strongly Connected Component*, *Biconnected Component*, *Global Minimum Cut*, *Max-flow Min-cut and Dijkstra's Shortest Path algorithm* and do numerical analysis of these results. Our study indicate that the citation graph formed by computer science literature are connected very well and its widespread connectivity doesn't depend on "hubs" and "authorities". The experimental results also show that the macroscopic structure of the citation graph is different from the macroscopic structure of Web graph which is Bow Tie model. Also, based on the citation graph built by querying *ResearchIndex* which is a subset and snapshot of whole citation graph, we provide the diameter measurements.

Contents

List of Figures

List of Tables

1 Introduction

Many natural systems form a huge dynamic directed graph, such that the nodes are the elements of the natural systems and the edges are the interactions among these elements. Researchers and scientists from different fields have common interests to explore the features hidden behind these huge directed graph. Since for a huge directed graph, the number of its nodes and edges usually is millions and billions. It is infeasible to construct a detailed directed graph structure to depict it. However, many researches have developed alternate methods to explore and utilize the characteristics of huge dynamic directed graphs. Particularly, the development of modern graph theory and graph-theoretic algorithms facilitates the study in these systems.

Developing and understanding huge dynamic networks greatly aid people in efficient and effective information location and knowledge discovery. For example, the citation graph formed by scientific literature whose nodes are articles and whose edges are links to articles cited in a paper conveys information about scholarly activities and spawns measures of scientific productivity. Intuitively, well-known papers tend to be cited frequently and papers dealing with the same specialty tend to connect to one another. There have been a number of scientific investigations on quantifying citations as measures of academic output. Nevertheless, interpreting link topology of the citation graph and offering insights into the nature of underlying inter-relationships such as those among people or specialties have not advanced greatly, and are still attractive and promising.

People in Information Science have studied the output of science for many years based on citation and co-citation analysis. Bibliometrics, Informetrics are the technical names for a range of analytical methods using publishing materials to develop statistics, multidimensional analyses. Yet the questions that what is the structure of citation graph? and what are its properties? still remained. Now things have changed. There are more and more published research papers available in WWW. The emergence of digital libraries gives us the chance to explore citation graph by employing modern graph-theoretic techniques. *ResearchIndex*[17], as one of such digital libraries, provides a very convenient way to do computer science article search and allows people to search relevant papers by navigating the links between papers formed by the citations, i.e., navigating along citation graph to find useful information. Given a such easily accessible digital library of computer science literature, we are eager to explore the Citation Graph of Computer Science Literature and mine its link structure for structural pattern discovery.

The following considerations motivated our study. Understanding the link topology of the citation graph using graph-theoretic tools may:

1. yield valuable insight into other citation or co-citation analyses.

2. facilitate knowledge discovery relying on link information such as similarity calculation, and finding communities.

3. help in citation graph visualization.

4. help evaluate the evolution of specialties or research themes over time.

Motivated, we constructed a web robot to query *ResearchIndex* to extract and build the citation graph autonomously. Due to the time and space limitation, the constructed citation graph is only a snapshot and subset of real citation graph of computer science literature. To this citation graph, we did three sets of experiments: 1. generated the in- and out- degree of nodes distributions, verifying that they follow the ubiquitous Power law distribution. 2. applied a series graph theoretic algorithms on this graph, checking its connectivity and finding various types of components. 3. using global minimum cut algorithm sliced this graph into pieces, exploring its interior structure. We will report our main results momentarily.

1.1 Main results

By constructing a robot for querying *ResearchIndex*, we built a collection of citation graphs. The results of our three sets of experiments on them indicated consistent characteristics existing among them:the first set of experiments of being to generate in- degree distributions showed that the in-degree numbers follow Power law distribution with 1.71 as exponent, i.e., the fraction of literature with k citations is proportional to $1/k^{1.71}$.

Our analysis of second set and third set of experiments, which are checking various types of components and applying global minimum cut algorithm, indicated that $\approx 90\%$ of all nodes form a single Weakly Connected Component(WCC) if citations are treated as undirected edges. In such giant WCC, almost 68.5% of the nodes have no any incoming link see Figure 1, suggesting that 68.5% of the publications have not been cited yet. Furthermore, in such giant WCC, around 58% of the nodes account for a big Biconnected Component(BCC), and almost all rest of nodes fall into trivial BCCs each of which consists of only one distinct node. Our analysis of the big BCC shows that there are 43% of its nodes without incoming link, and rest of its nodes have both incoming and outgoing links. Instead of being Bow Tie model as web graph, the macroscopic structure is a half Bow Tie with one side wing cut off (see Figure 2).

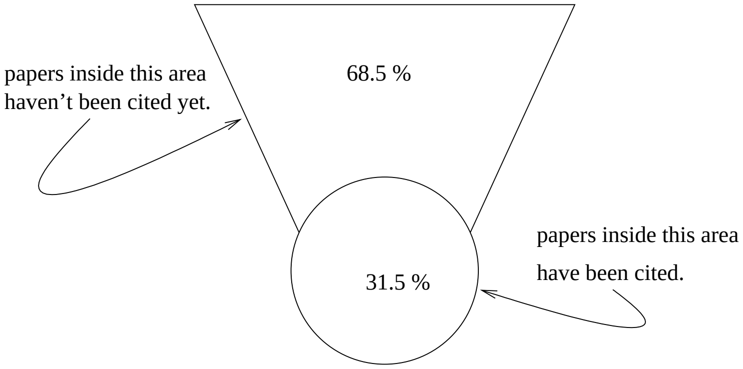

Figure 1: The connectivity of the citation graph: 68.5% of the nodes in WCC have no incoming link.

Treating the citations as directed edges in the citation graph, we showed that the directed diameter is 29. The diameter is defined as the maximum over all ordered pairs(u, v) of the shortest path from u to v. However, the probability of existing a directed path between any pair of nodes is only *2%*. The undirected diameter is 18 measured by ignoring the direction of edges.

The remainder of this thesis is organized as five additional sections and one appendix. Citation and co-citation analysis in scientific literature has been studied for decades, it is related to our work in some point. Other related work is the efforts of characterizing and mining other large scale networks such as web graph, we discuss them in Section 2. In Section 3, we give a description of the dataset which our study relies on. Graph theory terminology and algorithms as main tools for this study are reviewed in Section 4. Section 5 contains the bulk of our experiments and analysis of results. Finally, Section 6 contains our conclusion and discussion of future areas of research for the

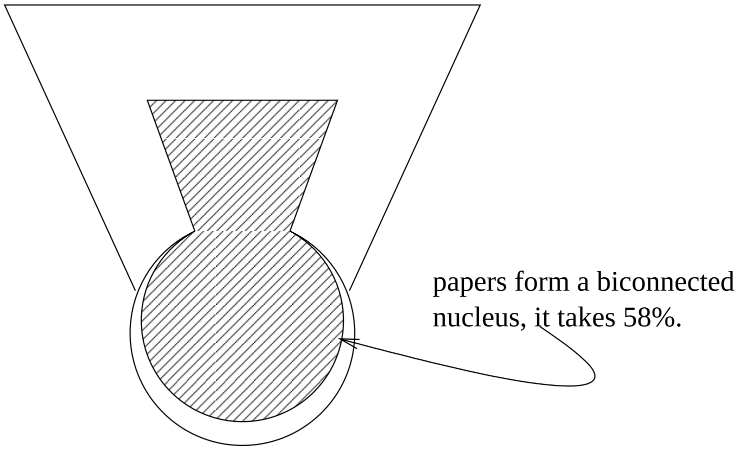

Figure 2: The connectivity of the citation graph: 58% of the nodes in the giant Weakly Connected Component(WCC) account for a big Biconnected Component(BCC). 43% of the nodes in the big BCC have no incoming link.

citation graph. We demonstrate our robot algorithm for data collection and discuss the consideration for building citation graphs from *ResearchIndex* in the appendix A.

2 Related work and literature review

Broadly speaking, related prior work can be divided into three categories:(1) citation and co-citation analysis on scientific literature, including visualization of citation network. (2) characterizing work on other large scale networks, including various measurements, and (3) efforts of mining link structure of networks for information location.

In this section, when we discuss the prior related work, we always make corresponding comparisons to our own study on the citation graph. The commonality consists in utilizing graph theory approaches for characterizing large scale networks as well as mining their link topology for efficient and effective knowledge discovery.

2.1 Citation and co-citation analyses of scientific literature

Many current approaches and algorithms for characterizing large scale networks such as World Wide Web, extend the research in the field of bibliometrics. Bibliometrics is the study of written literature and their citation structure. Someone might think that our study on citation graph of computer science literature is bibliometrics study on the citation of those literature. But actually, our work was inspired by work of characterizing Web graph. We are more interested in the graph structure formed by citations of literature instead of impact factor of each article. We consider our work is complementary to bibliometrics, and is a different view of citation structure. Research in bibliometrics has long been concerned with the use of citations to produce quantitative estimates of the importance and impact of individual scientific publication and journals. The most well-known measure in this field is Garfield's impact factor[10], used to provide computer-compiled statistical reports of Journal Citation Reports(JCR) of the Institute for Scientific Information(ISI). It is a measure of the frequency with which the average article in a journal has been cited in a particular year of period. It is observed that the impact factor is a ranking scheme based fundamentally on a pure counting of the in-degree of nodes in the citation graph, but it does not give us any picture of such graph. Bibliometrics can take a number of technical forms, as characterized as follows:(1) citation analysis: identify the number of times a specific publication is cited in other scientific publications, (2) co-citation analysis: identify pairs or groups of publications that are cited together in other publications, (3) co-word analysis: assign keywords to a publication by a professional reader; publications which have same keywords and sets of words are linked to each other via a clustering technique, and (4) scientific mapping: develop a visual model of the realm of scientific fields representing the structure of literature output of particular scientific fields.

Instead of analyzing the average or total number of citations such as impact factor, Redner[26] focuses on the more fundamental distribution of citations of scientific literature, namely, the number of papers which have been cited a total of x times, $N(x)$. The study in [26] is based on two relatively large data sets: one is the citation distribution of 783,339 papers during the period 1981-June 1997 that have been catalogued by the Institute for Scientific Information(ISI);the second is the citation distribution , as of June 1997, of the 24,296 papers cited at least once which were published in volumes 11 through 50 of Physical Review D(PRD), 1975-1994. Its focus is on citations of publications rather than citations of specific authors. The main result of this study is that the asymptotic tail of the citation distribution appears to be described by a Power law, $N(x) \sim x^{-\alpha}$, with $\alpha \approx 3$. Another important aspect of citation statistics is its continuing temporal evolution. This feature is nicely illustrated by the annual citation statistics of Physical Review D(PRD) publications, where the average number of citations for articles published in a given year is typically decreasing slowly with time. The citation distribution provides basic insights about the relative popularity of scientific publications and provides a much more complete measure of popularity than the average or total number of citations such as impact factor. [26] shows, at a basic level, most publications are minimally recognized, with $\approx 47\%$ of the papers in the Institute for Scientific Information(ISI) data set uncited, more than 80% cited 10 times or less, and $\approx 0.01\%$ cited more than 1000 times.

As scientific mapping, Chen [6, 7] develop a set of methods that extends and transforms traditional author co-citation analysis by extracting structural patterns from scientific literature and representing them in a 3D knowledge landscape. [6] address the problem to effectively and intuitively access and explore information in a digital library by a set of visualization tools. Their work focuses on two major datasets: the ACM SIGCHI conference series containing 169 papers published in three conference proceedings, and the ACM Hypertext conference series including all the papers published in the ACM Hypertext conference proceedings(1987-1998). In [7], the authors show their procedure for extracting intellectual structure from scientific literature. Their approach to knowledge visualization work particularly well for identifying intellectual groupings based on an extension of the traditional author co-citation analysis. Their results reveal many challenges for understanding knowledge structure, they argue that because citation analysis builds on scientists' long-established citation practice, approaches that focus on Web-based citation resource hold promise. Our work is based on such citation resource:*ResearchIndex*. As we listed above, the visualization approaches of extracting intellectual structure from scientific literature developed in [6, 7] give useful insights into understanding the structure of citation graph of scientific literature. Users can apply such visualizations to discover patterns and make valuable connections between articles. We consider that our work on characterizing citation graph of scientific literature will shed additional light on visualization of citation graphs.

2.2 Characteristics of large scale networks: Web graph

Consider the directed graph whose nodes correspond to static pages on the web, and whose edges correspond to hyperlinks between these pages. A.Broder and others in [3] study various properties of this graph including its diameter, degree distributions, connected components, and macroscopic structure. They performed a number of experiments on web crawls from May 1999 and October 1999–approximately 200 million pages and 1.5 billion hyperlinks. First, they verified the in- and out- degree distribution follow the Power law distribution with exponent as 2.1, confirming it as a basic web property. In their second set of experiments they studied the directed and undirected connected components of the web. Their analyses reveals an interesting picture of the web's macroscopic structure. Most (over 90%) of the nodes form a single connected component if hyperlinks are treated as undirected edges. This connected component breaks naturally into four pieces. The first piece is a central core, all of whose pages can reach one another along directed hyperlinks–this "giant strongly connected component" (SCC) is at the heart of the web. The second and third pieces are called IN and OUT. IN consists of pages that can reach SCC, but cannot be reached from it–possibly new sites that people have not yet discovered and linked to. OUT consists of pages that accessible from SCC, but don't link back to it, such as corporate websites containing only internal links. Finally, the TENTRILS contain pages that can not reach SCC, and cannot be reached from SCC. Perhaps the most surprising fact is that the size of SCC is relatively small– it comprises about $\frac{1}{4}$ of all pages. Each of other three sets contain about other three $\frac{1}{4}$ portions of all pages–thus, all four sets have roughly the same size. They call it as Bow Tie model.

Defining the diameter as the expected length of the shortest path where the expectation is over uniform choices from the set of all ordered pairs of nodes(u, v) such that there is a path from u to v, A.Broder et.al in [3] show that the diameter of the central core(SCC) is at least 28, and that the diameter of the graph as a whole is over 500. They show that for randomly chosen source and destination pages, the probability that any path exists from the source to the destination is only 24%. They also show that, if a directed path exists, its average length will be about 16, likewise, if a undirected path exists, its average length will be about 6.

The work of A.Broder et.al in [3] confirms the early work of Barabasi([23, 1, 2]) in which Barabasi introduce the scale-free characteristics of random networks–the probability $P(k)$ that a vertex in the network is connected to k other vertices decays as a Power law in some large random networks. Barabasi in [2] claim that the scale-free characteristics exist in many natural systems. In particular, many of these systems form complex networks. For example, living systems form a huge genetic network, whose vertices are proteins, the edges representing the chemical interactions between them. Similarly, a large network is formed by the nervous system, whose vertices are the nerve cells, connected by axons. But equally complex networks occur in social science, where vertices are individuals or organizations, and the edges characterize the interactions between them, in the business world, where vertices are companies and edges represent diverse trade relationships. In order to find the generic features of such network development, they explore the large database describing the topology of large network as WWW. To determine the local connectivity of the WWW, they constructed a robot. The data were obtained from the complete map of the *nd.edu* domain, that contains 325,719 documents and 1,469,680 links. From the collected data they determined the probability $P_{out}(k)(P_{in}(k))$ that a document has k outgoing(incoming) links follow a Power law, with 2.45(2.1) as exponent. Another particularly important quantity in a search process is the shortest path between two documents, d, defined as the smallest number of URL links that must be followed to navigate from one document to the other. They find that the average of d over all pairs of vertices is 19. Connecting to its scale-free stationary state of large random networks, we expect that citation graph exhibit the same scale-free state rather in different size; but the issue is that what is its exponent? Scale-free stationary state gives us such insight that we may wish to understand the link structure of citation graph by exploiting a subgraph, provided the subgraph has enough nodes and edges.

2.3 literature review on mining link structure of large scale networks for information location

Understanding the topology and local connectivity of large random networks allows us to predict the behavior of diverse algorithms for locating information and patterns in these networks. To my understand, searching information and patterns in Hypertext structures by exploiting link topology started as early as Botafogo in [5] in 1991. The authors define two types of important nodes:index and reference. An index node is a node whose out-degree is greater than average out-degree, a reference node is a node whose in-degree is greater than average in-degree. In order to better capture the notion of how complex a hypertext is, the *compactness* metric is developed. Informally, the compactness is measured by the distances between nodes. Giving the definition of compactness, they define a semantic cluster of a hypertext as a set of nodes and links that have two properties:(1) they are a subgraph of the hypertext, (b) the compactness of the subgraph is higher than the compactness of the whole graph. Having the definition of semantic cluster of hypertext, they introduce two types of algorithms to find semantic clusters in the hypertext:(1) Biconnected components , and (2) Strongly connected components. By analyzing the structure of a hypertext using both algorithms, [5] show that it is possible to identify groups of nodes that have a high semantic relation. In citation graph domain, we expect to find Biconnected component and Strongly connected component structures. The main insight brought by [5] is that could we capture the notion of publication community by compactness? and how to identify communities in the context of citation graph?

Many achievements have been fulfilled in exploiting link topology of Web to locate information in recent researches. Kleinberg[16] proposes an algorithm that, given a topic, finds pages that are considered authorities on that topic. The algorithm, known as HITS, is based on the hypothesis that for broad topics, authority is conferred by a set of hub pages, which are recursively defined as a set of pages with a large number of links to many relevant authorities. Specifically, their approach mainly address the problem of distilling and filtering authorities from large volume of relevant information. It consists of two processes: first, they need to construct a focused subgraph of WWW with these properties: (1) it's relatively small, (2) it rich in relevant pages, and (3) it contains most of the strongest authorities. Second, based on their hypothesis that there are hub pages which have links to multiple relevant authoritative pages;hubs and authorities exhibit what could be called a mutually reinforcing relationship:a good hub is a page that point to many good authorities;a good authority is a page that is pointed to by many good hubs, they apply a iterative algorithm to compute the hub weights and authority weight of each web page. They show that the vectors of hub and authority weights correspond to the principal eigenvectors of matrices inferred from the adjacency matrix of the focused subgraph of WWW. The main concerns are fundamentally different from problems of clustering. Clustering addresses the issue of dissecting a heterogeneous population into sub-populations that are in some way more cohesive; thus, clustering is intrinsically different from the issue of distilling authorities from a relevant corpus of broad topics. The hypothesis of hubs and authorities exhibiting mutually reinforcing relationship on which the HITS algorithm based is not likely to be expected in the context of scientific literature–citation graph. When [16] began with the goal of discovering authoritative pages, they are expecting mutually reinforcing relationship is the intrinsic property of WWW. On the other hand, their approach in fact identified a more complex pattern of social organization on the WWW, in which hub pages link densely to a set of thematically related authorities. This equilibrium between hubs and authorities is a phenomenon that recurs in the context of a wide variety of topics on the WWW. But in the context of scientific literature, it has typically lacked, and arguably not required, an analogous formulation of the role that hubs play in WWW. Therefore, when we explore citation graph of scientific literature, we have to avoid the pitfalls of using the notions of hub and authority. We argue that the framework of [16] seems appropriate as a model of the way in which authority is conferred by hubs in an environment such as the Web.

In addition to the two-level algorithm such as HITS in [16] to filter authorities from WWW, there have been several one-level approaches to ranking pages in the context of hypertext and the WWW. Brin and Page [25] proposed a ranking measure based on a node-to-node weight-propagation scheme and its analysis via eigenvectors. Their approach is based on a model in which authority is passed directly from authorities to other authorities, without interposing a notion of hub pages. Such model is more likely to be expected in the context of citation graph of scientific literature, therefore, we would like to say much about it here. They make use of link topology of the WWW to calculate a quality ranking for each web page; this ranking is called PageRank. They show that PageRank is an objective measure of its citation importance that corresponds well with people's subjective idea of importance; we would have the same feeling in citation graph of scientific literature. Analogous to scientific literature, they calculate page's importance or quality by counting the citations of a given page. Instead of counting the citation directly, PageRank extends this idea by not counting links from all pages equally. It is worth noting a basic concern in the application of this approach to WWW. The PageRank algorithm is applied to compute ranks for all the nodes in millions pages of the WWW; these ranks are then primarily used to order the results of subsequent text-based searches. In the context of citation graph of scientific literature, we more concern finding the activities of specialties instead of identifying ranks of all documents for a whole.

There are many approaches in knowledge discovery or similarity calculation based on link analyses of directed or undirected graphs representing the underlying objects. Rafiei and Mendelzon [22] consider a question of finding reputation of a given page by analyzing its neighborhood connection. Specifically, in the context of Web, they propose two methods for computing the reputation of web page in terms of random walks on the Web graph. Their first method is based on one-level weight propagation, PageRank, model proposed by Brin and Page [25]; whereas, their second method is based on two-level weight propagation, hubs and authorities model proposed by Kleinberg [16].

Kumar and others [24] show that a large number of Web communities can be identified from their signatures in the form of complete bipartite subgraph of the web based on the hub-and-authority structure of community proposed in [16]. Their main concerns are to find these implicitly defined communities in the Web. Specifically, they mainly focus on the co-citation relationship of webpages which occurs repeatedly. The main idea is that related pages are frequently referenced together, a phenomenon originated in the scientific literature. Their thesis is that co-citation is not just a characteristic of well-developed and explicitly-known communities but an early indicator of newly emerging communities. In other word, they can exploit co-citation in the web graph to extract all communities that have taken shape on the web, even before the participants have realized that they have formed a community. The process and results of trawling implicitly-defined web communities indicate that hub-and-authority structure is well appropriate in the environment such as WWW. In our citation graph of scientific literature domain, We also concern to find implicitly-define communities before participants have realized that they have formed such communities;these communities are early indicators of emerging specialties. The finding of emerging specialties would be help for researchers to justify their current research and identify their future research direction.

There are two more types of efforts in finding relevancy among web pages through exploiting link connection information of Web graph. One [8] such work is based on the hub-and-authority structure proposed by [16]; another [12] is based on the graph theoretic algorithm–Max-flow and Min-cut–to identify web communities defined in terms of connectivity among web pages.

Dean and Henzinger [8] describe the Companion and Cocitation algorithms, two algorithms which use only the hyperlink structure of the web to identify related web pages. Their Companion algorithm is derived from the HITS [16] algorithm, and their Cocitation algorithm finds pages that are frequently cocited with the input page(that is, it finds other pages that are pointed to by many other pages that all also point to the input page).

Flake, Lawrence and Giles in [12] propose an efficient approach to find Web communities by computing Min-cut through Max-flow on a derived underlying local web graph. They argue that if time and space complexity issues were irrelevant, then one could identify tightly coupled communities by solving the problem as a balanced minimum cut problem, where the goal is to partition a graph such that the edge weight between the partitions is minimized while maintaining partitions of a minimal size. But unfortunately, most generic versions of balanced minimum-cut graph partitioning are NP-complete. If balanced restriction is removed, the algorithm is easier. But one will suffer from the problem of highly unbalanced and trivial partitions in a graph. In order to avoid the problems mentioned above, they propose the approach to find communities combining Max flow-Min cuts algorithm with expectation maximization technique. They show that a community can be identified by calculating the $s - t$ minimum cut of graph with s and t being used as the source and sink, respectively.In our domain of citation graph of scientific literature, we notice that since we have a moderately size graph comparing to web graph, we would be able to identify research communities by balanced minimum cut graph partitioning. Actually, we define our notion of community analogous to web community defined in [12], and we propose a heuristic approach to solve the balanced minimum cut graph partitioning problem to identify communities in our context of the citation graph. We detail our approach in corresponding later section.

3 Description of dataset: digital library and autonomous citation index: *ResearchIndex*

ResearchIndex [15] is a Web-based digital library and citation database of scientific literature which are accessible from WWW. Traditionally, most published scientific literature appears in paper documents such as scholarly journals or conference proceedings. But with the WWW becoming an important distribution medium for scientific research, Web publications are often available, and they also eliminate the time lag between the completion of research and the availability of such publication in terms of paper documents. In order to assist the user in finding relevant Web based research publications, Bollacker, Lawrence and Giles[17] developed *CiteSeer*, an “assistant agent” which improves research paper searching. Their results produce the prominent digital library and citation database of Computer Science literature:*ResearchIndex*.

As a citation database, *ResearchIndex* keeps staying up-to-date with recently published articles. *ResearchIndex* should complement commercial citation indices such as the Institute for Scientific Information’s Science Citation Index(SCI). But in our study, we consider that *ResearchIndex* is sufficiently accurate and useful. A citation index catalogues the citations that an article makes, linking the articles with the cited works. Citation indices were originally designed mainly for information retrieval and to allow navigate the literature in unique ways, such as backward in time(through the list of cited articles) or forward in time(to find more recent, related articles). The availability of *ResearchIndex* provides the opportunity to us to build snapshots of citation graphs of computer science literature for our study by autonomously querying *ResearchIndex*.

As being the dataset of this study, *ResearchIndex* has its inherent shortcomings. First of all, *ResearchIndex* was created by a robot crawling the Web, the information of its database is that which is only accessible in the Web. Secondly, the same paper cited by different articles may appear in different formats, it may not appear in the database of *ResearchIndex* uniquely. Thirdly, the database of *ResearchIndex* only gathers works in the Web beyond a point in time, older papers may not present.

4 Terminology of graph theory and review of graph theoretic algorithms

In this section we briefly review the topics of graph theory and useful terminology as well as graph-theoretic algorithms. More details about graph theory could be found in [9] and in-depth development and implementation of graph-theoretic algorithms are described in detail in *LEDA* [19].

A *graph*(undirected) is a pair $G = (V, E)$ of sets satisfying $E \subseteq [V]^2$;thus the elements of E are 2-element subsets of V. The elements of V are the *vertices* of graph G, the elements of E are its *edges*.A graph with vertex set V is said to be a graph on v. The vertex set of a graph G is referred to as $V(G)$, its edge set as $E(G)$. The number of vertices of a graph G is its *order*. A vertex v is incident with an edge e if $v \in e$. Two vertices x, y of G are adjacent, or neighbors, if xy is an edge of G. Two edges $e \neq f$ are adjacent if they have an end in common. If all vertices of G are pairwise adjacent, then G is *complete*. A complete graph on n vertices is a K^n. Pairwise non-adjacent vertices or edges are called independent.

Let $G' = (V', E')$, if $V' \subseteq V$ and $E' \subseteq E$, then G' is a subgraph of G, written as $G' \subseteq G$. if $G' \subseteq G$ contains all the edges $xy \in E$ with $x, y \in V'$, then G' is an induced subgraph of G.

The *degree* of a vertex v is the number of edges incident at v, denoted as $d(v)$. A vertex of degree 0 is *isolated*. The number $\delta(G) = min\{d(v)|v \in V\}$ is the *minimum degree* of G, the number $\Delta(G) = max\{d(v)|v \in V\}$ is its *maximum degree*. If all the vertices of G have the same degree k, then G is *k-regular*. The number

$$d(G) = \frac{1}{|V|} \sum_{v \in V} d(v)$$

is the *average degree* of G.

A non-empty graph G is called connected if any two of its vertices are linked by a path in G. G is called *k-connected* if $|G| > k$ and $G - X$ is connected for every set $X \subseteq V$ with $|X| < k$. The greatest integer k such that G is k-connected is the *connectivity* $\kappa(G)$ of G. If $|G| > l$ and $G - F$ is connected for every set $F \subseteq E$ of fewer than l edges, then G is called $l - edge - connected$. The greatest integer l such that G is l-edge-connected is the *edge-connectivity* $\lambda(G)$ of G.

An *acyclic* graph, one not containing any cycles, is called a *forest*. A connected forest is called *tree*.

Let $r \geq 2$ be an integer, A graph $G = (V, E)$ is called *r-partite* if V admits a partition into r classes such that every edge has its ends in different classes: vertices in the same partition class must not be adjacent. When $r = 2$, it is called *bipartite*.

A *directed graph (or digraph)* is a pair (V, E) of disjoint sets (of vertices and edges) together with functions associating with each $e \in E$ a *source* $source(e) \in V$ and a *target* $target(e) \in V$. In other words, each edge has two *end* nodes, to which it is said to be *incident*, and a direction from *one(source)* to *another(target)*. The terminology and notation of digraph theory is similar to that of undirected graph theory. In fact, to every digraph there corresponds a graph, obtained by letting the edges to be the edges and ignoring the edge directions. Two edges of a digraph is *parallel* if they have the same source and target, and a digraph is *simple* is it has no loops or parallel edges. A digraph may be simple as a digraph, but not as a graph.

Given a digraph G, the *out-degree* of a vertex v is the number of edges incident v letting v as source;the *in-degree* of a vertex v is the number of edges incident v letting v as target.

When we write $e = vw, e \in E$ for an edge of G, we mean that $v = source(e), w = target(e)$.

An edge of a path $P = v_0, e_1, v_1, ..., e_k, v_k$ is *forward* if $source(e_i) = v_{i-1}$ and $target(e_i) = v_i$ and is *reverse* otherwise. A path in which every edge is forward is a *directed path* or *dipath*. A *directed cycle* is a dipath that is also a cycle.

Let $G = (V, E)$ be a directed graph and let v and w be two vertices of G. w is *reachable* from v if there is a path in G from v to w, i.e., if either $v = w$ or there is a sequence $e_1, ...e_k$ of edges of G with $k \geq 1$, $v = source(e_1), w = target(e_k)$, and $target(e_i) = source(e_{i+1})$ for all $i, 1 \leq i < k$.

A directed graph G is called *strongly connected* if from any node of G there is a path to any other node of G. A *Strongly Connected Component(SCC)* of a graph G is a maximal strongly connected subgraph. *LEDA*[19] implemented a procedure to compute Strongly Connected Component as:

```
int STRONG_COMPONENTS(const graph & G, node_array<int> & comp_num)
```

This procedure returns the number of strongly connected components of G and computes a node_array $< int >$ *comp_num* with encoding the strongly connected components of G. It runs in linear time $O(n + m)$, where $n = |V|$ and $m = |E|$.

Let $G = (V, E)$ be an undirected graph, A *Weakly Connected Component(WCC)* of G is a maximal connected subgraph of G. The procedure implemented in *LEDA*[19]

```
int COMPONENTS(const graph & G, node_array<int>& comp_num)
```

computes the number of connected components of G. It runs in linear time $O(n + m)$, where $n = |V|$ and $m = |E|$.

A connected undirected graph $G = (V, E)$ is called *biconnected* if $G - v$ is connected for every $v \in V$. Here

$$G - v = (V - v, \{e; e \in E \; and \; v \cap e = \emptyset\})$$

is the graph obtained by removing the vertex v and all edges incident to v from G. A *Biconnected Component(BCC)* is a maximal biconnected subgraph. A vertex v is called a *cutvertex* of G if $G - v$ is not connected. The procedure in *LEDA*[19]

```
int BICONNECTED_COMPONENTS(const graph& G, edge_array<int> &
comp_num)
```

returns the number of bccs of undirected version of G and the running time is $O(n + m)$.

Let $G = (V, E)$ be an undirected graph(self-loops and parallel edges are allowed) and let $w : E \rightarrow R_{\geq 0}$(R is real set) be a *non-negative* weight function on the edges of G. A cut C of G is any subset of V with $\phi \neq C \neq V$. The weight of a cut is the total weight of the edges crossing the cut, i.e.,

$$w(C) = \sum_{e \in E; |e \cap C| = 1} w(e)$$

A *minimum cut* is a cut of minimum weight. The function implemented in *LEDA* [19]

```
int MIN_CUT(const graph & G,const edge_array<int>& weight,
list<node> & C, bool use_heuristic=true)
```

takes a graph G and a weight function on the edges and computes a minimum cut. The running time of the algorithm is $O(nm + n^2 \log n)$.

Let $G = (V, E)$ be a directed graph, let s and t be distinct vertices in G and $cap : E \rightarrow R_{\geq 0}$(R is real set) be a non-negative function on the edges of G. For an edge e, we call $cap(e)$ the *capacity* of e. An $(s,t) - flow$ is a function $f : E \rightarrow R_{\geq 0}$ satisfying the capacity constrains and the conservation constrains:

(1) $0 \leq f(e) \leq cap(e)$ for $\forall e \in E$

(2) $\sum_{e;source(e)=v} f(e) = \sum_{e;target(e)=v} f(e)$ for $\forall v \in V \backslash \{s,t\}$

We call s and t the source and the sink of the flow problem, respectively. The value of a flow f, denoted $|f|$, is the excess of the sink, i.e.,

$$|f| = \sum_{e;source(e)=t} f(e) - \sum_{e;target(e)=t} f(e)$$

A flow is called *maximum*, if its value is at least as large as the value of any other flow. The function implemented in *LEDA*[19]

```
NT MAX_FLOW_T(const graph& G, node s, node t, const edge_array<NT>
& cap, edge_array<NT> & f)
```

computes a maximum flow f in the network (G, s, t, cap) and returns the value of the flow.

We close this section with the famous max-flow-min-cut theorem. An $(s,t) - cut$ is a set S of nodes with $s \in S$ and t not in S. The capacity of a cut is the total capacity of the edges leaving the cut, i.e.,

$$cap(S) = \sum_{e \in E \cap (S \times T)} cap(e)$$

Then

$$max\{|f| : f \ a \ (s,t) - flow\} = min\{cap(S) : S \ a \ (s,t) - cut\}$$

5 Characteristics of the Citation Graph of Computer Science Literature

It has been observed that a number of random networks spanning as diverse fields as the WWW or the people interaction social network exhibit consistent characteristics [2] that are independent of the nature of the system and the identity of its constituents. One such characteristic is that the in- and out- degrees follow Power law distributions. In addition, with the availability of large dataset of WWW, people have studied the linkage structure of web graph and developed approaches of exploiting link structure for information discovery. People's investigation of scientific citation numbers in Physics [26] only shows the existence of power law in degree distributions without exploring its citation graph. The availability of digital library and citation index enables us to build the citation graph formed by computer science literature as an example of huge networks with thousands of nodes and millions of edges. We now examine the Citation Graph of Computer Science Literature in greater detail. In this section, our study is answering questions such like: does the citation graph, as other huge networks display, exhibits the power law distribution existing in its degrees? how is the connectivity of the citation graph? can we find aggregate information of the citation graph by applying those developed efficient graph-theoretic algorithms? what does the macroscopic structure of the citation graph look like? Consequently, in this section, we not only verify its degree distributions, but also explore its connectivity, components and interior structure.

5.1 Building Citation Graph by querying *ResearchIndex*

The first step of characterizing the citation graph is building it from citation database. After constructing a robot, we built citation graphs by querying *ResearchIndex* autonomously. We choose three subtopics of computer science as start points to fulfill our task of building citation graph. The three topics are: *Neural Networks, Automata and Software Engineering*.

After we provide the topics to *ResearchIndex*, it will retrieve 1000-2000 papers related to each of these topics. For each topic, taking returned papers as a base set β, we start building citation graph CG through following procedure:

```
while $|CG| < predefined\,threshold$ and new paper is adding to $\beta$
        for each $v \in \beta$ has not been visited
             add $v$'s neighbors who point to $v$ to $\beta$
             add $v$'s neighbors to whom are pointed by $v$ to $\beta$
             mark $v$ visited
        let $CG$ be the graph induced by papers in $\beta$
end while
```

We have observed that there are two types of articles in *ResearchIndex*: one type of articles contributes fully information to citation graph, they are in *ResearchIndex*'s database; another type of articles only contributes half information to citation graph, i.e., we have no way to know who are their references, they were brought into *ResearchIndex* by references of other papers, but they are not in *ResearchIndex*'s database themselves.

After crawling for months, three citation graphs were built and one union citation graph was created by combining these three citation graphs. A preliminary analysis of the collection of citation graphs is shown in Table 1.

	(1)	(2)	(3)	(4)
	Number of papers be visited	Number of papers in database	Number of papers without citation over (2)	Number of citations of most-cited paper
citation graph–N.N.	109,519	23,371	16,555	739
citation graph–Automata	117,702	28,168	19,809	503
citation graph–S.E.	94,179	19,018	12,934	186
union citation graph	261,708	57,239	37,348	739

Table 1: The preliminary analysis of the collection of citation graphs.

5.2 Measurements of Citation Graphs of Computer Science Literature

In this section we describe empirical observations drawn from a number of our measurement experiments on the citation graphs obtained by querying *ResearchIndex* as above. The measurements include degree distribution results and diameters.

5.2.1 Degree distributions

We begin by considering the in-degree of nodes in the citation graph. "Distribution with an inverse polynomial tail have been observed in a number of contexts. The earliest observation is in the context of economic models [21]. Subsequently, these statistical behavior have been observed in context of literacy vocabulary [14], sociological models [13] etc"[1]. Most recently, people have observed that the degree distributions in web graph [2, 3] and the scientific citations [26] follow power law as well.

In our context of citation graph of computer science literature, we also observed that the in-degree distributions follow a power law: the fraction of papers with in-degree i is proportional to $1/i^\gamma$ for some $\gamma > 1$.

Our empirical experiments on all three citation graphs built from different topics as well as the union citation graph confirmed this result at a variety of scales. In all these experiments, the value of the exponent γ in the power law for in-degrees is a remarkably consistent 1.71.

Figure 3 is a log-log plot of the in-degree distribution of the union citation graph. The tail end of the distribution is 'messy' - there are only a few papers with a large number of citations. For example, the most cited papers had 739 citations, but the next most cited papers had 639 citations. It might be tempted to fit the curve in Figure 3 to a line to extract the exponent γ. However, there are so few data points in that range, simply fitting a straight line to the data would give not good slope. To get the proper fit, we need to bin the data into exponentially wider bins as shown in Figure 4. The value $\gamma = 1.71$ is derived from the slope of the line providing the best linear fit to the data in the figure.

The out-degree of a paper in our citation graphs depends on the age of the paper, since older papers will have fewer references that are in the database. Therefore, these out-degrees do not give an accurate picture of the out-degrees of nodes in the complete citation graphs, and therefore their distribution has not been considered here.

[1] see [3]

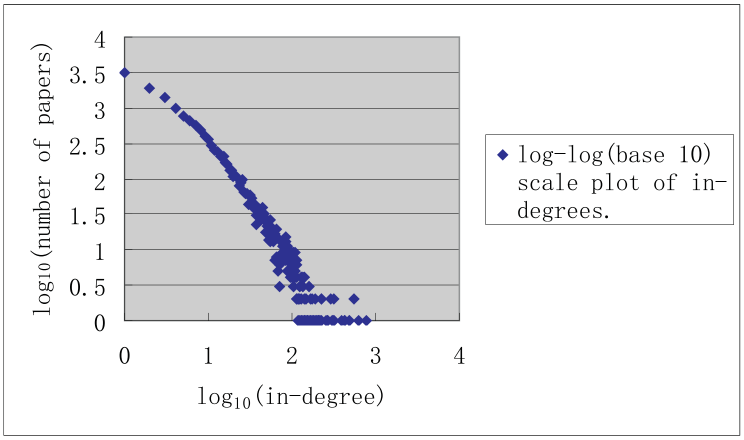

Figure 3: The in-degree distribution in the union citation graph in computer science literature subscribe to the power law.

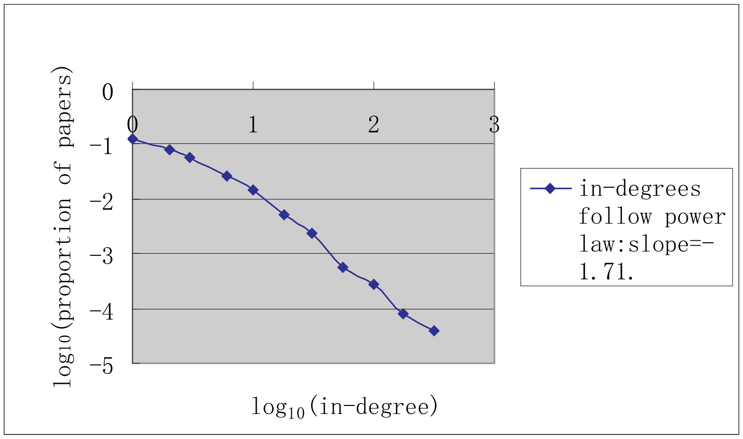

Figure 4: To bin the in-degree data into exponentially wider bins in the union citation graph in computer science literature:it subscribes to the power law with exponent=1.71.

5.2.2 Diameters

We turn next to the diameter measurement of citation graphs. In this study, the diameter is defined as the maximum over all ordered pairs(u, v) of the shortest path from u to v in citation graph. We measured two types of diameters of citation graph: directed diameter and undirected diameter. Directed diameter is measured by directed shortest path or dipath, while undirected diameter is obtained by treating edges without direction.

Before we measure diameters, we tested the connectivity of citation graph in terms of undirected graph. The results revealed that the citation graph is not connected. However, $\approx 80\% - 90\%$ of the nodes are in one connected component, while the rest form a few very small components. Details are described in Section 5.3. Focusing on connected component, we measure the diameters upon the giant connected component of citation graph.

Applying Dijkstra's shortest path algorithm on the giant connected components of citation graphs built from three different topics and union graph, we show the diameters in different graphs in Table 2.

	graph size	directed diameter	undirected diameter
citation graph–N.N.	23,371	24	18
citation graph–Automata	28,168	33	19
citation graph–S.E.	19,018	22	16
union citation graph	57,239	37	19
average		29	18

Table 2: The diameters of citation graphs built from different topics as well as union citation graph. Topic: N.N.: Neural Networks, S.E.: Software Engineering.

Ignoring the orientation of edges in citation graph, we observed that the citation graph is a 'small world', the undirected diameter is around 18, consistent at variety of scales and topics. In contrast, we don't have such 'small world' in *directed* citation graph. Our statistical study shows that the probability of existing a directed path between any pair of nodes is only *2%*, even though the measured directed diameter is around 30.

5.3 Reachability and components

We now consider the connectivity of Citation Graph of Computer Science Literature, involving examining the various types of connected components and reachability of nodes. Given a citation graph $G = (V, E)$, we will view G as a directed graph as well as undirected graph by ignoring the direction of all edges. We now ask how well-connected the citation graph is. Its connectivity can be examined in terms of both directed version and undirected version of G. For the undirected version of G, we ask: is the citation graph connected? what is its biconnectivity? For the directed version of G, we make crucial use of the orientation of edges: is the citation graph strongly connected?

We apply a set of algorithms that compute reachability information and structural information of directed and undirected citation graphs: *Weakly Connected Component(WCC), Strongly Connected Component(SCC) and Biconnected Component(BCC).*

As we mentioned before, we created three subgraphs formed by articles related to three different topics and one union graph. These raw graphs contain both nodes in *ResearchIndex*'s database as well as nodes not in *ResearchIndex*' database. We cleaned up raw graphs by discarding nodes who are not in *ResearchIndex*'s database, keeping only the nodes that contribute full information to citation graphs. After clean up, the graph sizes become: subgraph coming from neural networks papers contains 23,371 nodes; subgraph coming from automata papers contains 28,167 nodes; subgraph coming from software engineering contains 19,017 nodes; union graph contains 57,238 nodes (see Table 1). Our connected component experiments are applied to those cleaned up graphs.

5.3.1 Weakly connected components

Mathematically, a *Weakly Connected Component(WCC)* of undirected graph $G = (V, E)$ is a maximal connected subgraph of G. A WCC of a citation graph is a set of articles each of which is reachable from any other if links may be followed either forwards or backwards. In the context of a citation graph, links stand for the citations from one article to other articles cited in the former one. The WCC structure of a citation graph gives us an aggregate picture of groups of articles that are loosely related to each other.

The results drawn from the weakly connected component experiments on citation graphs are shown in Table 3. The results reveal that the citation graph is well connected–a significant constant fraction $\approx 80\% - 90\%$ of all nodes fall into one giant connected component. It is remarkable that the same general results on connectivity are observed in each of the three topic subgraphs. In turn, the same behavior is observed for the union graph, suggesting a certain degree of self-similarity.

	graph size	size of largest WCC	percentage of largest WCC	size of second largest WCC
citation graph–N.N.	23,371	18,603	79.6%	21
citation graph–Automata	28,168	25,922	92%	20
citation graph–S.E.	19,018	16,723	87.9%	12
union citation graph	57,239	50,228	87.8%	21

Table 3: The results of Weakly Connected Component experiments on different citation graphs: the majority ($\approx 90\%$) of articles are connected to each other if links are treated as without directions.citation graph:N.N stands for Neural Networks; S.E. stands for Software Engineering.

Derived from the results of WCC experiments, a picture represents the connected component of the citation graph is shown in Figure 5.

5.3.2 Strongly connected components

We turn next to the extraction of *Strongly Connected Component(SCC)* of the connected components of the three topical citation graphs and their union graph. A *Strongly Connected Component(SCC)* of a directed graph is a maximal subgraph such that for all pairs of vertices (u, v) of the subgraph, there exists a directed path (dipath) from u to v. In the context of the citation graph, a dipath from u to v means that article u directly cites article v or article u cites a intermediate article w, w cites next intermediate article and so on, until it reaches article v indirectly. Since there is a temporal direction between citing article and cited article, if article u directly or indirectly cites article v, then v would not cite back to u. As a result, we might expect that there is no SCC in the citation graph.

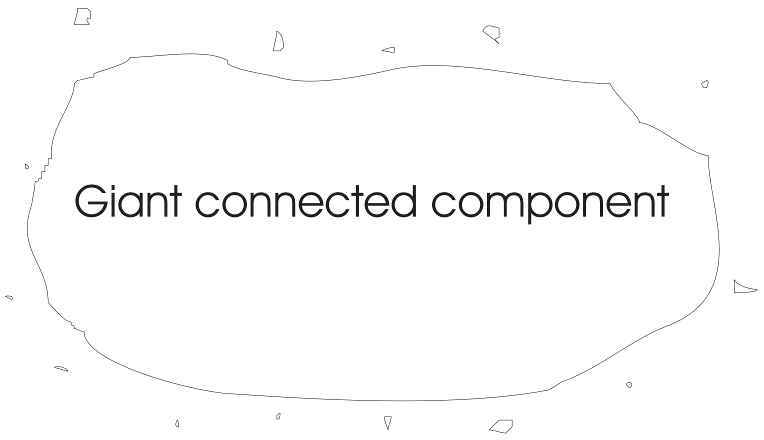

Figure 5: The connectivity of the citation graph: around 90% of the nodes fall into a giant connected component, the rest forms a several very small components.

But contrary to our expectation, the results of SCC experiments on the collection of citation graphs reveal that there exist one to three sizable SCC's in each of the citation graphs, as well as a few very small SCC's. The results drawn from the experiments are shown in Table 4.

	graph size	size of largest SCC	size of second largest SCC	size of third largest SCC
citation graph–N.N.	18,603	144	14	10
citation graph–Automata	25,922	192	29	24
citation graph–S.E.	16,723	17	11	8
union citation graph	50,228	239	155	60

Table 4: The results of Strongly Connected Component experiments on different citation graphs: there exist many small SCCs, among them there are one -three bigger SCC(s), the rest are even smaller comparing those bigger ones. citation graph:N.N stands for Neural Networks; S.E. stands for Software Engineering.

In order to know which publications formed the SCCs, i.e. how the directed cycles were generated in those citation graphs, we extracted some SCCs from citation graphs and searched articles of these SCCs directly in *ResearchIndex*'s database to find their titles, abstracts, authors, journals and published years. Our study shows that several types of publications formed SCCs: (1) publications written by same authors tend to cite each other, they usually produce self-citations, (2) publications which are tightly relevant tend to cite each other, e.g., publications, whose authors in same institute, dealing with same specialty and getting published concurrently are highly relevant and tend to cite each other, (3) publication which got published in different publication types such as journals, inproceedings or technical reports in different time formed directed cycles with other publications. Such a publication was considered as the same one during our creation process of citation graph. (4) books or other publications which got published in several editions in different time often acted as jump points in citation graph. Since different editions of publication were treated as the same one node in citation graph, acting as jump point forming directed cycles in citation graph; that is the reason of the existence of one to three bigger SCCs, while (1) - (3) types of articles often fell into even smaller SCCs containing 2-5 articles. For example, the paper entitled “Option Decision Trees with Majority Votes(1997)” by R. Kohavi appearing on “Machine Learning:Proceedings of the Fourteenth International Conference” and the paper entitled “Data Mining using MLC++ – A Machine Learning Library in C++(1997)” by R. Kohavi and others appearing on “Tools with Artificial Intelligence” are cited by each other. One more example is, there are three papers who are paper1 entitled “Efficient Distribution-free Learning of Probabilistic Concepts(1994)” by M.J.Kearns et al. appearing on “Computational Learning Theory and Natural Learning Systems, Volume I”, paper2 entitled “Toward Efficient Agnostic Learning(1992)” by M.J.Kearns et al. appearing on “In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory” and paper3 entitled “Learning Switching Concepts(1992)” by A. Blum appearing on “COLT:Proceedings of the Workshop on Computational Learning Theory” forming a directed cycle on which paper1 cites paper2, paper2 cites paper3 and paper3 cites the earlier appearance of paper1 on “In Proceedings of the Thirty-First Annual Symposium on Foundation of Computer Science(1990)”.

A conceptual map that is elicited from analysis of results of SCC experiment on the union citation graph is depicted in Figure 6. A number of small SCCs are embedded in a well connected background net. This background net is a directed acyclic structure, i.e., there is no directed cycle in background net.

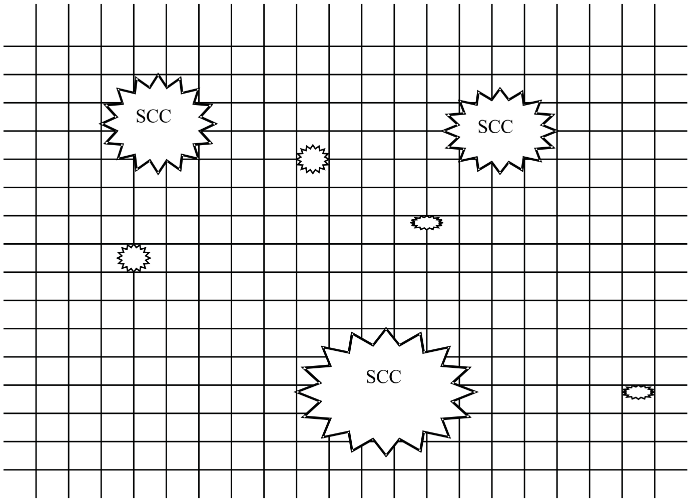

Figure 6: The directed connectivity of the citation graph: a number of small SCCs embedded in a background net;the background net is a directed acyclic graph.

5.3.3 Biconnected components

We are not satisfied with the coarse structural information drawn from WCC experiments on the undirected version of the citation graph such that there exists a unique giant connected component.

However, our effort of further refining the structure of the citation graph by looking at SCC which makes use of orientation of edges revealed that this giant connected component does not generally contain any large strongly connected subgraphs–we couldn't obtain explicit aggregate information in terms of directed version of the citation graph. Consequently, we turn our focus to stronger connectivity measure in terms of undirected graph–*biconnectivity*.

A Biconnected Component(BCC) in the citation graph is a set of nodes such that two nodes u and v are biconnected if there is no third node w so that w lies on all $u - v$ paths if links may be followed either forwards or backwards. Applying the biconnected component algorithm on the giant connected components of citation graphs, we find that each giant connected component of each citation graph contains a giant biconnected component. The giant biconnected component acts as a central biconnected nucleus, with small pieces connected to this nucleus by cutvertices, and other single trivial nodes connected to this nucleus or some small pieces. A coarse tree structure in terms of biconnected components is a bipartite, $H = (A \cup B, E)$, where A is the set of cutvertices and B is the set of its BCCs. Thus, we can intuitively picture the structure of citation graph as Figure 7.

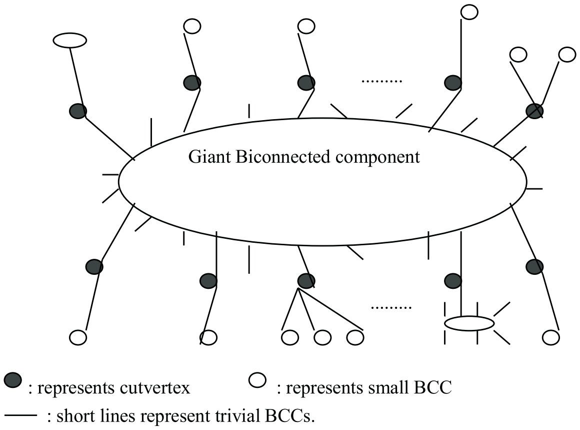
Giant Biconnected component
....
....
● : represents cutvertex
○ : represents small BCC
—— : short lines represent trivial BCCs.

Figure 7: The biconnectivity of the citation graph:a giant BCC acts as nucleus, with small pieces connected to it by cutvertices, and other single trivial nodes connected to nucleus or some small pieces. Such a bipartite is formed by a set of cutvertices and a set of BCCs.

The numerical analysis of sizes of BCCs indicated that $\approx 58\%$ of all nodes account for the giant biconnected nucleus, the rest $\approx 40\%$ of all nodes are in trivial BCCs each of which contains single distinct node, remaining $\approx 2\%$ of all nodes fall into a few small pieces. A histogram of size analysis is depicted in Figure 8. Our analysis of the big BCC shows that there are 43% of its nodes without incoming link, and rest of its nodes have both incoming and outgoing links.

5.3.4 Aggregate picture

By performing a set of connected component algorithms, we are able to elicit an aggregate picture of the citation graph as an undirected graph. Our analysis of WCC experiment indicates that $\approx 90\%$ of the nodes form a giant Weakly Connected Component(WCC); such a single giant WCC can be

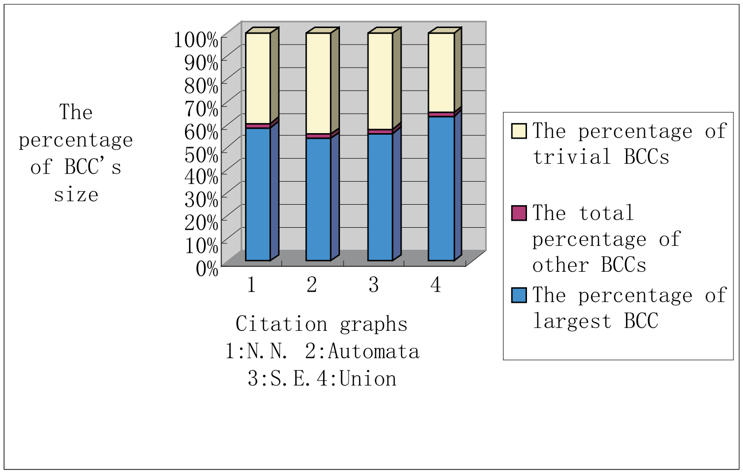

Figure 8: Analysis of sizes of BCCs: average 58% of the nodes form the giant biconnected nucleus, average 40% of the nodes are in trivial BCCs connected to other non-trivial BCCs, average 2% of the nodes fall into a few small pieces. Citation graph:N.N. stands for Neural Networks, S.E. stands for Software Engineering, Union stands for Union citation graph.

divided into two parts: one part contains almost 68.5% of the nodes without any incoming link, suggesting that 68.5% of publications have not been cited yet, another part contains the rest of publications with at least one citation. Finally, in such a giant WCC, around 58% of nodes form a big Biconnected Component(BCC) act as a biconnected nucleus, with a few small BCCs connected to this nucleus by cut vertices, and all rest of nodes fall into trivial BCCs each of which consists of single distinct node connected to this nucleus or some other small pieces. The aggregate picture is shown in Figure 1 and Figure 2.

5.4 Does connectivity depend on some key articles?

We have observed that the citation graph is well connected–90% of the nodes in a giant connected component containing another biconnected nucleus, 58% of the nodes, if we treat citation graph as undirected graph. The result that the in-degree distributions follow the Power law indicates that there are a few nodes of large in-degree. We are interested in determining whether the widespread connectivity of the citation graph results from a few nodes of large in-degree acting as “authorities”. As to the out-degree, since we clean up the raw citation graph by discarding nodes without complete contributions to the citation graph, the out-degree of the kept nodes does not actually represent the number of references of the corresponding papers. Yet we are still interested in knowing whether the connectivity of the citation graph depends on nodes with large out-degree acting as “hubs” in the citation graph. We test this connectivity by removing those nodes with large in-degree or out-degree, and computing again the size of the largest WCC. The results are shown in Table 5 and Table 6.

Testing connectivity of union citation graph

Surprisingly, the results show that the widespread connectivity does not depend on “hubs” and

size of graph	50,228							
k	200	150	100	50	10	5	4	3
size of graph after removing	50,222	50,215	50,152	49,775	46,850	43,962	42,969	41,246
size of largest WCC	50,107	49,990	48,973	43,073	26,098	14,677	9,963	1,140

Table 5: Sizes of the largest Weakly Connected Components(WCCs) when nodes with in-degree at least k are removed from the giant connected component of union citation graph

size of graph	50,228							
k	200	150	100	50	10	5	4	3
size of graph after removing	50,225	50,225	50,224	50,205	48,061	43,964	42,238	39,622
size of largest WCC	50,202	50,202	50,198	50,131	46,092	37,556	33,279	26,489

Table 6: Sizes of the largest Weakly Connected Components(WCCs) when nodes with out-degree at least k are removed from the giant connected component of union citation graph

“authorities”. Indeed, even if all links to nodes with in-degree 5 or higher are removed(certainly including links to every well-known articles on computer science), the graph still contains a giant Weakly Connected Component(WCC). Similarly, if all links to nodes with out-degree 3 or higher are removed, the graph is still well connected. In order to measure how is the graph connected after removing “authorities ” and “hubs”, two histograms are obtained, representing the percentage of the giant WCC over the graph after removing “authorities” and “hubs” in Figure 9 and Figure 10.

The analysis of sizes of giant WCCs indicate that “authorities” have more heavy influence on connectivity than “hubs”, relatively. Since even nodes with 3 out-degree are removed, there still are more than 60% nodes falling in a giant WCC; in contrast, when “authorities” with in-degree 3 are removed, the graph has a great number of isolated components. Our observations drawn from widespread connectivity tests have two interesting aspects: first, the connectivity of citation graph is extremely resilient and does not due to the existence of “hubs” and “authorities”; second, “hubs” and “authorities” are embedded in a graph that is well connected without their contributions.

5.5 Minimum cuts

A question related to understanding the structure of the citation graph is how to find thematically cohesive communities. So far, our study of various types of connected components has resulted in a well-connected citation graph with a giant biconnected nucleus. The next question is whether there is any further structure within that nucleus. We attack this problem using minimum cut algorithms, both for global minimum cut and for minimum cuts between specific pairs of nodes.

Mathematically, an (edge) cut C of graph $H = (V, E)$ is a set of edges which, when removed, disconnect the graph. The size of a cut is the number of edges in the cut. Given an edge weight function $w : E \to R$, a minimum cut is a cut whose total weight is minimum.

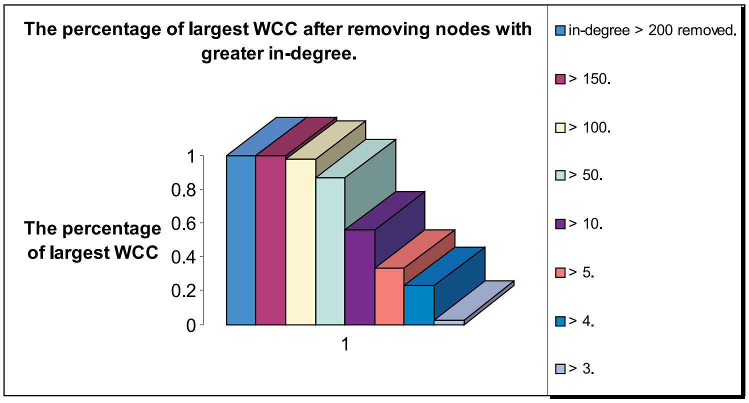

Figure 9: Analysis of sizes of giant WCC after removing "authorities": the height of each bar indicates the percentage of the largest WCC of the graph after removing nodes with $k = 200, 150, 100, 50, 10, 5, 4, 3$ in-degree.

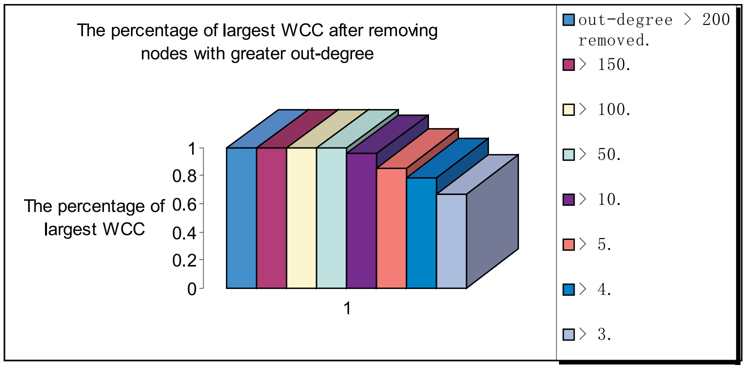

Figure 10: Analysis of sizes of giant WCC after removing "hubs": the height of each bar indicates the percentage of the largest WCC of the graph after removing nodes with $k = 200, 150, 100, 50, 10, 5, 4, 3$ out-degree

Our min-cut experiments focus on the giant connected component of union citation graph. After extracting the giant connected component $H = (V, E)$ from the union citation graph, we assign edge weight $w(e) = 1, for\ all\ e \in E$, and we apply the global minimum cut algorithm on H. In order to explore the interior structure of graph H, we implemented the min-cut exploring procedure as below:

```
procedure Explore_Min_Cut($H = (V, E)$)
    while $|H| > 0$
        compute min-cut $C$ of $H$
        calculate edge weight over crossing edge set $F$
        let $H_1 = (C, E_1)$ be graph induced by $C$
        let $H_2 = (V - C, E_2)$ be graph induced by $V - C$
        Explore_Min_Cut($H_1$)
        Explore_Min_Cut($H_2$)
    end while
```

Most of the resulting cuts of the above procedure reveals are trivial cuts, each of which separates only one node from the rest of the graph.

When the procedure *Explore_Min_Cut(H)* is applied to graph H recursively until the fragments become trivially small, we find that the percentage of trivial cuts is 99%.

From these experiments we conclude that the interior link structure of the citation graph is dense; also that there is *no* such explicit community information discernible through global minimum cuts, and we need more sophisticated tools for finding communities in citation graph.

Since the global minimum cut approach does not give us communities, we turn to the computation of minimum cuts between specific nodes. If two nodes are selected to belong to different topics, then possibly the minimum cut between them might separate the papers belonging to the two topics. To investigate this hypothesis we selected authority papers (papers with large in-degree) belonging to the topics of Neural Networks, Automata and Software Engineering, and we computed the minimum cuts of the union graph between pairs of such papers. For the minimum cut computation, we made two modifications to the (directed) union graph of 50,228 nodes:

1. We add the reverse edges to all the edges of the graph, so as to effectively treat it as an undirected graph.

2. We add node capacities equal to 1, by the following construction. We replace each node v by two nodes v_{in} and v_{out} and an edge from v_{in} to v_{out}, and we connect all the edges into v to v_{in} and all the edges out of v to v_{out}. All edges have capacity 1, thus allowing us to associate capacities 1 to all the nodes, as well.

The resulting graph has 100,456 nodes.

From these experiments we obtain highly unbalanced partitions of the union graph. The cut sizes are similar to the in-degree of the nodes, and the smaller partition contains at most a few hundred nodes while the larger partition contains the rest of the nodes (approximately 99,000).

6 Conclusion and future work

We now make conclusion of our work and indicate the revealed challenges that future work needs to resolve.

6.1 Conclusion

To the extent of traditional scientific citation analysis, our study of characterizing the citation graph of computer science literature facilitates understanding the intellectual structures in computer science based on citation index of computer science literature. First of all, the degree distributions provide basic insights about the relative popularity of publications in computer science. At a basic level, most publications are minimally recognized– only 1/3 of publications are cited, and 83% of those cited publications are cited 10 times or less. Publication with citations greater than 10 is relatively rarer. Secondly, being ignored the direction of links representing the citations from one article to others, the citation graph is well connected, manifesting a ‘small world’. However, it is worthwhile to note that the probability of existing a directed path between any pair of nodes is only 2%, if the links may only be followed forwards. Thirdly, our approaches of structural analysis of citation graph augment the traditional citation co-citation analysis of scientific literature. Particularly, the integration of various citation patterns and the graph theoretic analysis provides a rich representation of a knowledge domain. People can apply such structural analysis to discover patterns and make valuable connections between publications or authors.

6.2 Future work

There a number of interesting further directions suggested by this study. First, the citation of a scientific article is a function of time, suggesting that the structure of citation graph is dynamic instead of static. In real life, the reputation of paper is fading with time. But not all papers are being known and forgotten in same time function. Intuitively, some well-known papers are getting more and more citations, then replaced by newly emerging papers in same specialty; some papers have not been cited yet, probably never; some papers drew attention in a certain period of time, then forgotten quickly. All these varieties of citations affects the structure of citation graph over time. We have not taken time factor into consideration in our study of characterizing. There are a number of interesting and fundamental questions that can be asked about the evolution of citation graph, involving both evolution of in-degree of individual articles and evolution of link topology of the graph. With the assistance of citation indices and digital libraries, we can easily build citation graphs with time stamp and individual nodes with time stamps too. Applying same graph-theoretic algorithm as in this study, we can obtained more statistical measures relating to characteristics of citation graph, such that the life expectancy of computer science articles, age distributions. Moreover, it challenges us to develop more sophisticated tools to study the evolution of local link structure of citation graph for predicting the research trends. Also, we can study the life span of specialties and communities, helping the researchers and scientists to predict their research outputs.

Second, there is a challenge of “hubs” and “authorities” analysis in citation graph. We have reviewed the related work for analyzing “hubs” and “authorities” in web graph in section 2. Although we have claimed that scientific citation graph and web graph are governed by different principles, and the equilibrium between “hubs” and “authorities” is an appropriate model for certain environment such as WWW, we do notice that there are a number of ‘survey’ and ‘review’ papers existing in many specialties of computer science field. Kleinberg [16] has made a comprehensive comparison of scientific citation standing and hub/authority measure in WWW. Our question is whether to ignore the existence of ‘survey’ or ‘review’ papers acting as “hubs” in scientific citation. Since most work in measuring the scientific citation is based on one-level model such as the method extended by PageRank [25], we expect that such two-level [16] model as hub/authority will give us insights to identify authorities in scientific citation graph. All ‘survey’ and ‘review’ papers acting as “hubs” in citation graph emerged in a certain time when a specialty has developed to a certain stage. Combining the evolution of graph discussed above, we can study the changes of citations of papers cited by “hubs” after survey or review. It will shed more lights on life spans of specialties.

Third, community identification and similarity calculation are still interesting problems to study. People have developed many other approaches purely exploiting link topology to find communities or compute similarity in web graph, involving hub/authority model[8], max flow-min cut[12]. The hub/authority model focuses on distill authorities from a corpus of tons of relevant pages, while max flow-min cut has mainly been used in focused crawling. Both are efficient and effective in the sense of online replying. In the case of identifying specialties and predicting research trends in long-term sense, we believe that graph decomposition and partitioning are appropriate models. It should be pointed out that the balanced graph partitioning problem is NP-hard [20], but local search strategies have been proposed for them [18, 11]. The strategy of assigning weights to edges for different relationships between nodes is still not fully understood in accurate level.

Finally, we believe that characterizing the link topology of scientific citation graph has the potential for beneficial overlap with a number of areas. one of these areas is the field of information retrieval. On the other hand, combining textural content of individual nodes as well as link topology of citation graph leads us to a promising future of knowledge discovery in the domain of scientific literature. One direction is how can we annotate autonomously the communities discovered by graph partitioning process without human intervention? we expect that efficiently and effectively discovering research patterns and filtering finer topics from broader range will boost the computer scientists’ theoretical and practical research activities.

References

[1] A-L.Barabasi and R.Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

[2] A-L.Barabasi, R.Albert, and H.Jeong. Scale-free characteristics of random networks:the topology of the world-wide web. *Physica A*, 281:69–77, 2000.

[3] A.Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. *Graph Structure in the Web*. IBM Corporation, http://www.almaden.ibm.com/cs/k53/www9.html, 2000.

[4] Alf-Christian Achilles. *The Collection of Computer Science Bibliographies*. University of Karlsrube, http://liinwww.ira.uka.de/bibliography/index.html, 2000.

[5] Rodrigo A. Botafogo and Ben Shneiderman. Identifying aggregates in hypertext structures. In *UK Conference on Hypertext*, pages 63–74, 1991.

[6] Chaomei Chen. Visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing and Management*, 35:401–420, 1999.

[7] Chaomei Chen. Visualising a knowledge domain's intellectual structure. *Computer*, 34:65–71, 2001.

[8] Jeffrey Dean and Monika Rauch Henzinger. Finding related pages in the world wide web. *WWW8/Computer Networks*, 31:1467–1479, 1999.

[9] Reinhard Diestel. *Graph Theory*. Springer, Springer-Verlag New York, 2000.

[10] E.Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178:471–479, 1972.

[11] Charles M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th IEEE Design Automation Conference*, pages 175–181, 1982.

[12] Gary William Flake, Steve Lawrence, and C.Lee Giles. Efficient identification of web communities. In *Proc. of ACM SIGKDD-2000*, pages 150–160, 2000.

[13] G.K.Zpif. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.

[14] G.U.Yule. *Statistical Study of Literacy Vocabulary*. Cambridge University Press, 1944.

[15] NEC Research Institute. *RearchIndex*. http://citeseer.nj.nec.com, 2001.

[16] J.M.Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[17] K.D.Bollacker, S.Lawrence, and C.L.Giles. Citeseer:an autonomous web agent for automatic retrieval and identification of interesting publications. In *Proceedings of the 2nd International Conference on Autonomous Agents,ACM*, pages 116–123, 1998.

[18] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 29(2):291–307, 1970.

[19] K.Mehlhorn and St.Naher. *LEDA*. Cambridge University press, http://www.mpi-sb.mpg.de/ mehlhorn/, 1999.

[20] M.Garey et al. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[21] V. Pareto. *Cours d'economie politique*. Lausanne et Paris, Rouge, 1897.

[22] Davood Rafiei and Alberto O.Mendelzon. What is this page known for? computing web page reputations. In *Proc. WWW9 Conference*, 2000.

[23] R.Albert, H.Jeong, and A-L.Barabasi. Diameter of the world-wide web. *Nature*, 401:130–131, September 1999.

[24] R.Kumar, P.Raghavan, S.Rajagopalan, and A.Tomkins. Trawling the web for emerging cyber-communities. *WWW8/Computer Networks*, 31:1481–1493, 1999.

[25] S.Brin and L.Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of the WWW7 Conference*, pages 107–117, 1998.

[26] S.Redner. How popular is your paper?an empirical study of the citation distribution. *European Physical Journal B*, 4:131–134, 1998.

A The citation graph in *ResearchIndex*

A.1 Article distributions in *ResearchIndex*

We apply a set of graph-theoretic algorithms on the citation graph to explore its link topology. There are several commercial citation indices such as ISI which contains many science fields in a single citation index. These citation indices of many science fields couple with each other so that to characterize citation graph in one special field becomes difficult by querying ISI citation indices. Moreover, the term format in ISI is more appropriate to researches in information science. Our goals of study mainly focus on exploring the linkage structure of Citation Graph of Computer Science Literature. We observed that the *ResearchIndex* is an appropriate data resource for our study. We have introduce *ResearchIndex* as a digital library of computer science literature in Section 3. *ResearchIndex* contains almost all publications available across WWW and their citations in computer science specialty. In this appendix, we introduce the considerations of querying *ResearchIndex* to create citation graph and algorithms of our robot. Before we construct any robot to query *ResearchIndex*, we have to understand the operation mechanisms of *ResearchIndex*. We need to answer several questions as below:

1. What is the format of searching result of *ResearchIndex* after submitting a query term?
2. How to effectively and efficiently perform the Breadth-First Search in *ResearchIndex* to get citation graph?
3. What does the citation graph roughly look like in *ResearchIndex*?

The collection of computer science bibliographies[4] gives us the following statistical data in Table 7 about the literature in computer science by Oct. 2000:

How many publications in *ResearchIndex*? In its homepage [15], it says that it contains around 300,000 documents and 4 million citations. But not all documents' full-text and full-references are available in *ResearchIndex*'s database. *ResearchIndex* contains two types of documents: one type is that its full text had been downloaded into database of *ResearchIndex* and its full references had got parsed; one type is that only its citation information had been parsed from the first type of documents and its full text is not available in the database of *ResearchIndex*. That means there are two types of nodes in citation graph after querying *ResearchIndex*: one type of nodes have complete contribution to citation graph; another type of nodes only have citation information without reference information, they have no complete contribution to citation graph. When the robot query the database of *ResearchIndex*, it will retrieve both types of nodes, so it influence the effectiveness of results of characterizing the citation graph.
Let's define:
Complete node in citation graph:The retrieved node in citation graph has its complete incoming links as well as outgoing links. In other words, if we start a certain search algorithm from this type of node, we can reach its children as well as its parents.
End node in citation graph:The retrieved node in citation graph only has its incoming links from other nodes, its outgoing links are lost. In other words, we couldn't follow this type node to reach its children.
Let's guess how many complete nodes in *ResearchIndex* can be used to characterize the citation graph:
(1) Using query topic: Neural Networks:

Subject Area	Journal Articles	Conference Papers	Technical Report	All Entries
Others/Unclassifieds	155307	119198	13306	321341
Theory/Foundations of CS	80255	34895	4828	135005
Mathematics	76043	5195	5795	99407
Artificial Intelligence	29460	29868	7405	90685
Parallel Processing	25930	28248	8717	75351
Computer Graphics	32649	20417	1555	61637
Technical Reports	303	467	53480	55235
Compiler	22003	10511	4805	48990
Softw.Eng./Formal Methods	17686	21363	2616	47607
Distributed Systems	18458	5199	2056	33163
Databases	11017	12387	2209	30607
Neural Networks	10605	6847	1470	23083
Human-Comp. Interaction	5628	12150	17	20316
Operating Systems	9017	6611	564	18672
Typesetting	4769	1365	170	9368
Logic Programming	1993	4946	583	8648
Object-Oriented	1382	3048	939	7133
Wavelets	1338	381	353	2867
Total	503843	323096	110868	1089115

Table 7: Statistics for the computer science bibliography collection

When we supply the query term “Neural Networks” to *ResearchIndex*, it replies the result as:’More than 10,000 results found,only retrieve 2000..’.*ResearchIndex* gives the information of whether this paper is in its database. Making a statistics on retrieved 1500 papers(Since the server is overloaded, the *ResearchIndex* only given 1500 results), we observed that 246 papers are in its database–complete nodes, and 1254 papers are not in its database—end nodes.Therefore, we could estimate that only 16.4% papers are complete nodes in citation graph related to the topic “Neural Networks”, 83.6% papers are end nodes in citation graph related to the topic “Neural Networks”.

(2) Using query topic:Information Retrieval:

When we supply the query term “Information Retrieval” to *ResearchIndex*, it found more than 10,000 papers, but only retrieved 1000 papers due to the overloading of server. 175 papers are in its database, 825 papers are not in database. Therefore, 17.5% papers are complete nodes and 82.5% papers are end nodes in citation graph related the topic “Information Retrieval”.

From the statistics’ point of view, we estimated that the citation graph in *ResearchIndex* has 17% complete nodes, other 83% nodes are end nodes. From this conclusion, we have to be careful when we explore the citation graph, since the end nodes don’t make complete contributions to citation graph.

The robot would query *ResearchIndex* to obtain the citation graph using forward Breadth-First Search, following the outgoing links of a set of start nodes obtained using topic searching. Since we have estimated that there are 17% nodes are complete nodes, we can follow their outgoing links to crawl the citation graph, but there will be a bunch of nodes only have outgoing links–in other words, they may be new publications haven’t been cited by others yet(see Figure11).

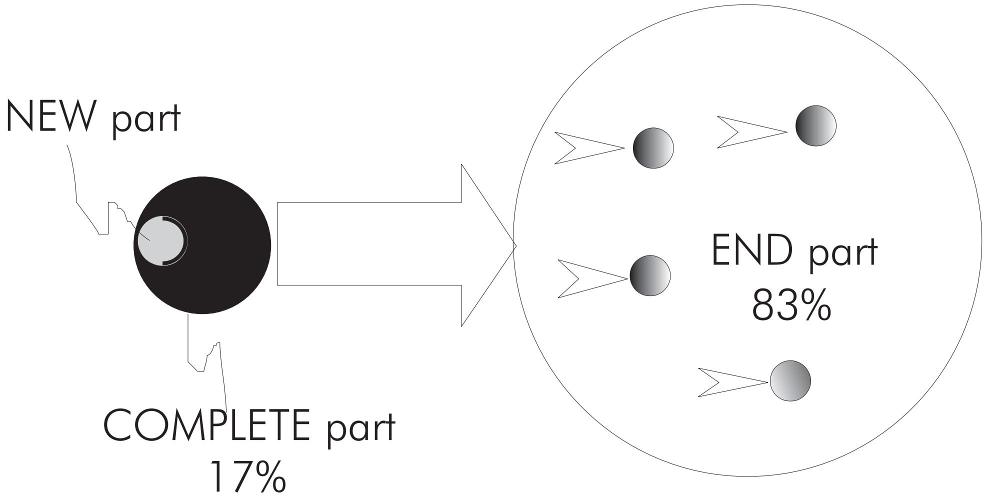

Figure 11: The macroscopic shape of citation graph in *ResearchIndex*: 83% nodes only have incoming links from others–they are end nodes–call them END part. 17% nodes not only have incoming links, but also have outgoing links—they are complete nodes–call them COMPLETE part. A fraction of COMPLETE part only have outgoing links–those new papers haven’t been cited by others–call them NEW part.

The BFS algorithm would probably lose them since they are no incoming links from others. To deal with this problem, there are two ways: first, we try to obtain as many start nodes as possible to perform forward BFS algorithm. Second, not only keep the outgoing adjacent list of each node, but also keep the incoming adjacent list of each node. we perform the BFS not only forwards, but also backwards to create the citation graph as completely as possible.

A.2 Considerations on querying *ResearchIndex*

Although there are many crawlers for crawling web pages, querying *ResearchIndex* to create the citation graph has some special issues. We need some special considerations for crawling the database of *ResearchIndex*.

A.2.1 Make sure that the citation graph is accurate

When we crawl the database of *ResearchIndex* for building the citation graph, we start from a set of nodes obtained by topic querying such that supplying the topic term "Neural Networks" to *ResearchIndex* to get more than thousands of links to the database of *ResearchIndex*. Those thousands of links form a FIFO queue of crawler, one link represents a node in citation graph.
NOTE:

1. How to compare two articles are exact same one during the BFS crawling process?

2. Need to update the incoming and outgoing adjacent list of encountered nodes during the BFS crawling process.

3. How to deal with the timeout event when BFS is processing a child of a node after having updated the adjacency list of some incoming neighbors and outgoing neighbors of this node? How to roll back? otherwise the citation graph is not accurate.

The *ResearchIndex* provides two types of replied webpapes when retrieve citations for a specific paper: Context page and DOC page. The context page provides the information such that: 1. How many times this paper have been cited by others—incoming links. 2. A fraction of papers which citing this paper. 3. The most important part:its bibentry. Since that two papers have the exact same bibentry is unlikely to happen, we use the bibentries of papers to compare them to decide whether they are the same one. The DOC page provides the references of this paper, therefore, we need to extract this important information and follow them to get its children and update its outgoing adjacency list.

When forward BFS starts, crawler opens context pages to get the bibentry for the purpose to compare papers. If this retrieved article hasn't been processed, open its doc page to extract its children. For each child, crawler needs to open each context page of this child to get its bibentry and compare to other retrieved papers and update its adjacent lists. When timeout event happens during processing, we have to roll back or restore its links in the end of queue for processing later.

A.3 High level design of robot for querying *ResearchIndex*

The crawler which is constructed for querying *ResearchIndex* to build the citation graph has to be able to deal with those questions mentioned in section A.2.1. The object diagram is in Figure12.

From the object diagram, it can be found that there are several main classes:

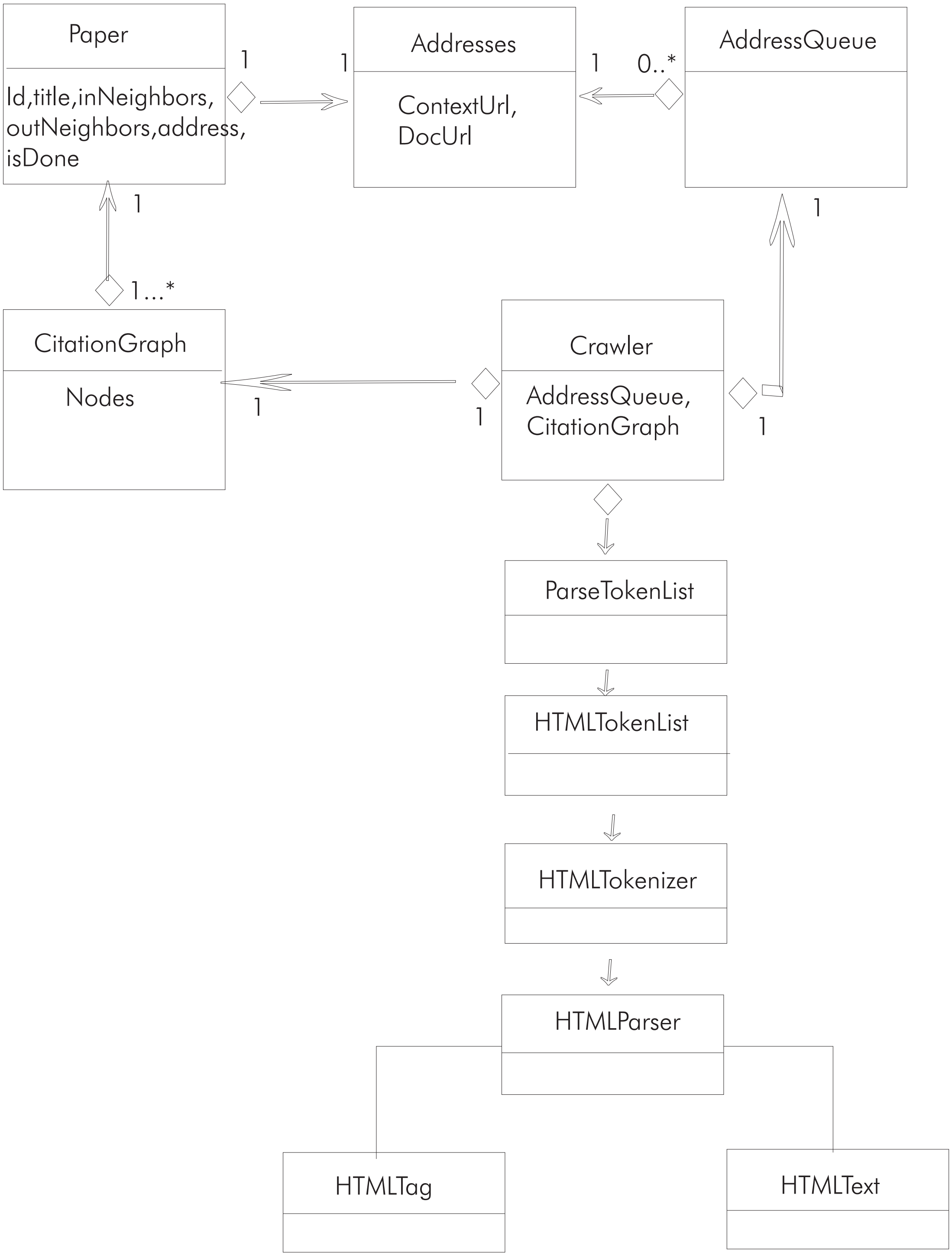

Figure 12: Object diagram of crawler for querying *ResearchIndex*:Using UML(Unified Modeling Language) notations.

1. class Paper:the node of citation graph holds its properties such as title,adjacent lists.
2. class Addresses:holds the context url and document url in *ResearchIndex*.
3. class Neighbors:adjacent list avoiding duplicate items.
4. class CitationGraph:citation graph stored in Hashtable for efficiently random access.
5. class ParseTokenList:utility class to parse the answers of *ResearchIndex*.
6. class Http:be able to deal with the timeout of socket connection with the *ResearchIndex* server.
7. Other help classes: AddressQueue, HTMLToken, HTMLTokenList, HTMLTag, HTMLText, HTMLTokenizer, HTMLParser etc.

The flow chart of crawler as Figure13

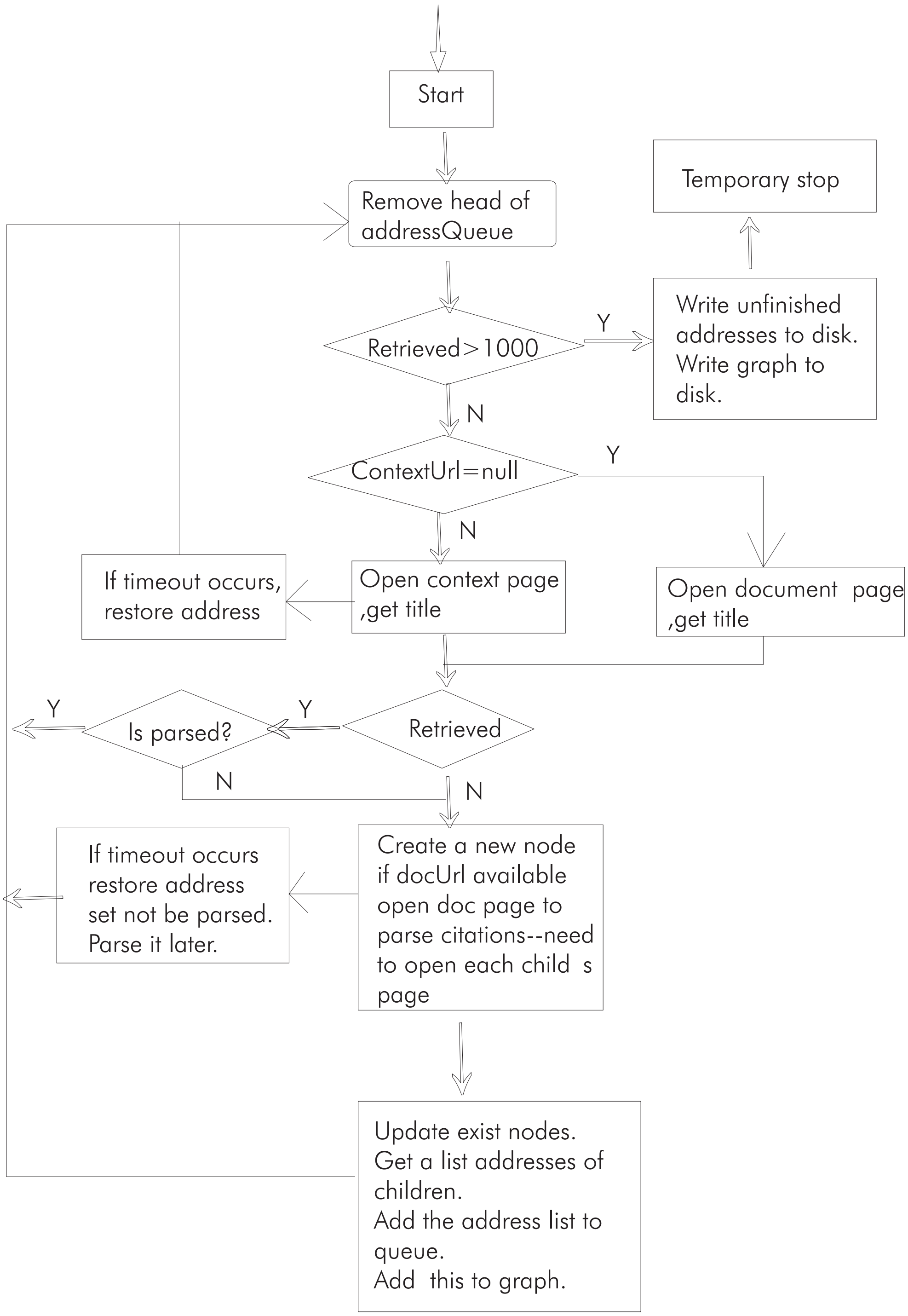
Start
Remove head of addressQueue
Temporary stop
Retrieved>1000
Y
Write unfinished addresses to disk. Write graph to disk.
N
ContextUrl=null
Y
N
If timeout occurs, restore address
Open context page ,get title
Open document page ,get title
Y
Is parsed?
Y
Retrieved
N
N
If timeout occurs restore address set not be parsed. Parse it later.
Create a new node if docUrl available open doc page to parse citations--need to open each child s page
Update exist nodes. Get a list addresses of children. Add the address list to queue. Add this to graph.

Figure 13: The flow chart of crawler for querying *ResearchIndex*.

A.4 Algorithm of robot for querying *ResearchIndex*

```
1.  Initialize the addressQueue of papers from disk;
2.  Initialize the citationGraph from disk;
3.  While addressQueue is not empty do
4.        if number of retrieved papers greater than a threshold
5.        Break;
6.        Pop the head of addressQueue;
7.        If context URL is not empty do
8.             Open context page;
9.             Deal with timeout;
10.             Get title;
11.             If it was retrieved
12.                 If it was processed
13.                     Go to 6;
14.        Open document page;
15.        Construct a new node;
16.        Process children page;
17.        Update neighbors’ adjacent lists;
18.        Deal with timeout;
19.        Add children’s addresses to addressQueue;
20.        Add this node to graph;
21.  End while;
22.  Write unfinished addresses to disk;
23.  Write graph to disk;
```