월드와이드웹의 내용기반 구조최적화

이우기* · 김 승** · 김한도*** · 강석호**

Optimization Model on the World Wide Web Organization with respect to Content Centric Measures

Wookey Lee* · Seung Kim** · Hando Kim*** · Sukho Kang**

Abstract

The structure of a Web site can prevent the search robots or crawling agents from confusion in the midst of huge forest of the Web pages. We formalize the view on the World Wide Web and generalize it as a hierarchy of Web objects such as the Web as a set of Web sites, and a Web site as a directed graph with Web nodes and Web edges. Our approach results in the optimal hierarchical structure that can maximize the weight, tf-idf (term frequency and inverse document frequency), that is one of the most widely accepted content centric measures in the information retrieval community, so that the measure can be used to embody the semantics of search query. The experimental results represent that the optimization model is an effective alternative in the dynamically changing Web environment by replacing conventional heuristic approaches.

Keyword : Web Structure Optimization, Web Graph, Integer Programming, Content Centric Weight Measure

1. Introduction

Structuring the World Wide Web yields significant insights into web algorithms for searching, discovering, mining, and revealing Web information. Explosively, growing number of Web pages requires a generic framework [1, 12, 17] that can provide a logical surrogate for the users as well as for search agents. The Web can be viewed as a digraph consisting of a set of Web

본 논문은 2004년도 한국경영과학회 추계학술대회(2004년 10월 23일)최우수 논문상(이론부문)수상논문으로 소정의 심사결과를 거쳐 게재 추천되었음.
* 성결대학교 컴퓨터공학부
** 서울대학교 산업공학과
*** KEDCOM Co. Ltd.

sites that have an initial node called homepage and many other Web nodes, where the Web nodes correspond to HTML files having page contents and the Web edges correspond to hypertext links interconnected with the Web nodes (which will be stated in section 2). The Web as a graph approach, however, is in reality too complex for Web snoopers or search robots to be guided in the forest of the whole WWW.

A tree architecture such as Web catalogues or a site map [5, 7, 20] is one of the typical examples of Web structuring. Naive static Web abstractions, however, do little to help a Web designer who wants to model a Web site and also often causes navigation problems of their own. The problem of finding a tree structure of a Web site from a Web graph having local cycles is known to NP-hard [12, 15]. When the Web site can be represented as a hierarchical structure, problems such as multiple paths, recursive cycle, multi-path cycle, and multiple parents should be resolved primarily [4, 14].

The stack oriented depth first search approaches are popular to adopt, because it has several strong points : from a cognitive science point of view, it's search methods is similar to the behaviors of human snoopers [12, 22]. The algorithms can run in linear time, since the running time of each step is proportional to the number of nodes removed from the stack at that step, and each node is detected at most once [9, 18]. The approach, however, to coin the Web structure has a severe weakness in the Web environment, because normally Web pages are complicatedly inter-connected with other Web pages, which can cause a long series of Web pages. This in turn entails the number of clicks and long time consumption to access each page.

On the other hand, the breadth first search algorithms have some different advantages. With this algorithm, an important Web page can easily be accessed by simply clicking relatively fewer steps from its homepage [10, 16]. The algorithms crawl most important Web pages that have many links and those links will be found early, regardless of on which host or page the crawl originates [2, 10]. It is easy to reduce a graph to a hierarchical structure and the depths for accessing each Web page. Some experimental researches have reported that graphical representations support better navigation because this type of representation more accurately matches a user's mental model of the system [8, 9, 22]. A hierarchical structure by the breadth first search is simple and well known to implement. It, however, is inappropriate for finding a significant page in a Web site or clustering the Web pages with semantics. Therefore, in this paper, we introduce a property called the "weight" to evaluate the significance of Web pages.

A semantic structuring algorithm [21] converts a web site structure to an unbiased tree, which minimizes an average distance from the root node to a specific Web page and resolves ties between semantic weights. They also considered the semantic relevance between the Web nodes. It has potentials in clustering and measuring personalized Web pages. The weakness of the approach is that it is too static so that even when a minor change in the Web page's weight or in the link structure occurs, then the entire structure needs to be reorganized. Corresponding to the modification of the Web structure, there are some intriguing findings that most pairs of pages on the web are separated by a handful of links, almost always under 20, and that this number will grow logarithmically with the size of the web [1]. We need a precise and robust

model to formalize our view on the Web with respect to a mathematical approach that can convert the Web graph to a hierarchical structure. In effect, the structure of a web site can prevent the search robots or crawling agents from confusion in the midst of the huge Web pages in the Web sites.

This paper is organized as follows. In Section 2, we present the data model of Web sites and the Web schema. In Section 3, we discuss keyword-based weight measure endowed on Web node. In Section 4, we will discuss the integer programming model of the Web structure. In Section 5, we will present an example of our model and the robustness of our model. In Section 6, we will discuss the test system called AnchorWoman which we developed. Finally, we will end with a conclusion.

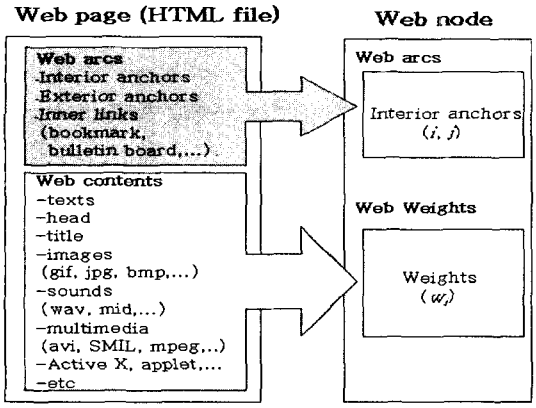
2. The Web Schema

A Web site is defined as a set of Web nodes $N_i = \{N_1, \cdots, N_n\}$, a directed graph $G_w = (N_w, E_w)$, an edge function $x_{ij} : N^k \rightarrow \{0, 1\}$, and $\forall(i, j) \in E_w$ consisting of a finite Web node set N_w, and a finite Web edge set E_w of ordered pairs of Web nodes, and the Web edge elements (i, j) respectively, where i, j = {0, 1, 2, 3, ···, n-1}, and n represents the cardinality of web pages = $|N_w|$. There is a natural mapping of the nodes that correspond to Web pages and the edges to Uniform Resource Identifiers [3, 10]. The Web node (N_W) can be defined as follows :

$$N_W = [N_i, \{(i, j), \forall i\}, w_i] \quad (1)$$

Where the N_i represents a node corresponding to an HTML file (we simply represent the Web node identifier as i), where the homepage is defined as a default page (index.html) is predetermined by the Web server. The $\{(i, j), \forall i\}$ is the set of Web edges having hypertext links from Web node N_i to N_j. The w_i represents the values specified by the measure of keywords (which will be discussed later in more detail). The Web page contents can be described as the attributes of the Web page such as title, Meta, format, size, modified date, text, figures, multimedia files, etc. For convenience, the weight in this paper is assumed to represent the Web page contents generated by the method described in Section 3. The Web node schema is represented in [Figuer 1]. A hierarchical abstraction is useful in organizing information and reducing the number of alternatives that must be considered at any one time [18].

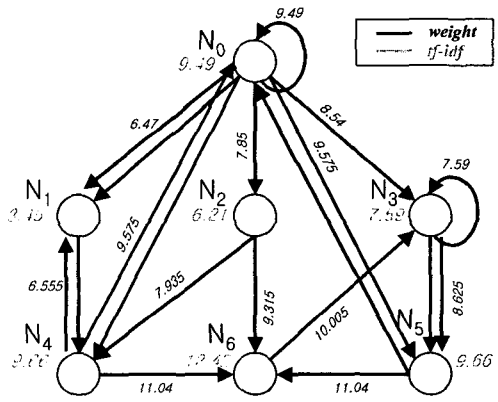
The URI's are classified into two types, i.e., interior edges and exterior edges [21] : the interior edges are the URI's that indicate the HTML files somewhere within the Web site ; while the exterior edges indicate the HTML files outside of the Web site. We are interested in the interior edges only, for we are focusing on the structure of a Web site from which the WWW can be represented as a digraph. After preprocessing the URI's of a Web site, the standard (full length) IP addresses of every Web page are derived. The exterior edges are, however, discarded in the preprocessing phase, because they have a different server IP address, i.e., a different site. Actually, in some web sites, there is only a frame in the default page (ex, index.html). In this case, we give the URI's of the Web pages which is included in frame. By the same reason, the implementation issues like redirecting in terms of cgi.bin or asp, Java applet are discarded.



[Figure 1] Web Schema Generation from the Web pages

3. Content Measure

Most Web ranking algorithms utilize a similarity measure in terms of the vector space model (VSM), which has been extensively studied in the information retrieval community. To compute the similarities among a set of Web pages, each page can be viewed as an n dimensional vector $<w_1, \cdots, w_m>^T$. A common way of computing a Web page's weight W is the *tf-idf*, which is obtained as an unnormalized vector $W' = < w'_1, \cdots, w'_m >^T$, where each w'_i is the product of a term frequency factor and an inverse document frequency factor. The *tf* factor is proportional to the frequency of the i^{th} word within a web page. The *idf* factor one divided by the number of times the word appears in the entire set of web pages that corresponds to the content discriminating power of the i^{th} word that appears rarely in documents has a high *idf*, while a word that occurs in a large number of documents has a low *idf*. Typically, *idf* is computed by logarithms from the total number of documents and is the number of documents containing the word. If a word appears in every document, its discriminating power is 0. If a word appears in a single document, its discriminating power can be very large. Once the unnormalized vector is computed, the normalized vector W is typically obtained by their norms. The similarity between a query Q and a web page Ni can then be defined as the inner product of the Q and N vectors. The weight specified in this paper indicates the importance of the Web page that indicates how statistically important it is [2, 6, 11, 19]. Note that our approach does not restrict the content similarity measures. The similarity measure in this paper is to measure the weight of Web node corresponding to their significance by the weight. We introduce the *tf-idf* as the weight measure which can be used to determine the topological ordering of Web sites. Then, simply by comparing the numerical differences of the *tf-idf*, it can be said that a node is closer to a specific node. As described previous, the *tf-idf* measure is applied as a weight of the Web node.



[Figure 2] Test Web site. The circle represents a Web node id and the arrow represents hyperlink. The nodes and links have weights.

The prototype system which is called Anchor Woman (ver. 2.0) has been tested, experiment to search for the structure of the test web site [23]. The link structure of the site is shown in refers [Figure 2].

4. Optimal Web Organization

4.1 Optimization Modeling for the Web

In the mathematical model, the variable x_{ij} is either 1 when an edge from node i to node j exists, or 0 else. The parameter w_{ij} represents an average weight from node i to node j. There are several alternatives for deriving the weight of a node and to generate a geometric distance to the link, including the number of inward or outward links [11]. In this paper, we use a weight as *tf-idf* to each node, and generate a Euclidean distance w_{ij} from node i to node j. It should be noted that our approach to construct a high level architecture for the Web, of course, is not constrained by the measure. The optimization model to maximize edge weights subject to several constraints are suggested as follows.

$$Max \sum_{j \in N} w_{ij} x_{ij} \quad (3)\text{-}1$$

$$s.t.\ \ Max \sum_{j \in N} x_{ij} = 0 \ \text{ if } \ j = 0 \ \ \forall\, i \quad (3)\text{-}2$$

$$\sum_{j \in N} x_{ij} \le 0 \ \text{ if } \ j \neq 0 \ \ \forall\, i \quad (3)\text{-}2'$$

$$x_{ij} + x_{ji} \le 1 \ \ \forall\, i, j \quad (3)\text{-}3$$

$$x_{ij_1} + \sum_{k=1}^{m-1} x_{j_k j_{k+1}} x_{j_m i} \le m-1 \ \text{ for } 2 \le m \le |N| \quad (3)\text{-}3'$$

$$x_{ii} = 0 \ \ \forall\, i \in N \quad (3)\text{-}4$$

$$x_{ij} = 0 \ \ \forall\, i, j \neq N \quad (3)\text{-}5$$

$$x_{ij} = 0 \ \textit{or}\ 1 \quad (3)\text{-}6$$

The objective function 3-(1) means the maximization of tree path's total sum of average weight. The constraint 3-(2) and 3-(2)' means each tree node's *indegree* should be 1 (except for the root node). The constraints 3-(3) and 3-(3)' are to remove a cycle and the constraints 3-(4) is to remove a self-cycle. The constraint 3-(6) represents the problem is a kind of Integer Programming. That is to say, whether there is a link or not. According to the constraint 3-(6), the variable x_{ij} can be 0 or 1, i.e. it can be used to remove multiple paths. Additionally, a virtual path from all nodes to all nodes without physical links should be nullified in constraint 3-(5). Finally, by using the above IP, the path that owns a high weight in the digraph would survive in the result tree.

4.2 Sub-modules for Organizing the Web

The Integer Programming model can be applied to transform the Web digraph into a tree. First, except for the root node, the tree node's indegree should be 1. Second, All cycle which may exist in a graph should be removed during the transformation phase. Cycle detection algorithm will be explained later. Third, a self-cycle within a graph node should be removed as in the following algorithm Cycle_Detection, too. So the duplicate paths between adjacent two nodes in a graph should be removed.

The cycle detection algorithm plays the role of detecting cycles in a digraph, and makes topological order of node in the digraph. It is also used to make *restriction 3-(3)*. Let a directed graph a directed graph $G_w = (N_w, E_w)$ have n node set N_w and m edge set E_w, and A(i) an adjacency list. The *indegree* is the number of Web edges that link into a Web page, the *outdegree* is the number of Web edges that link out from the Web page. The LIST is a data structure that stores

```
AlgorithmCycle_Detection
input
    Gw(Nw, Ew) a directed web graph, where Nw : set of web nodes, Ew : set of web edges
    A(i) = adjacency list
Output
    cycle link List
procedure
{
    for (all i, Nw) indegree(i) = 0 do
    for (all (i, j) ∈Ew) indegree(j) = indegree(j)+1 do
    LIST = next = 0 ;
    for (all i ∈ Nw) if (indegree(i) = 0), LIST = LIST{i} ;
    while (LIST){
                select a node i from LIST and delete it ;
                next = next + 1 ;
                order(i) = next ;
                for (all (i,j), A(i)) {
                        indegree(j) = indegree(j)-1 ;
                        if (indegree(j) = 0) LIST = LIST{j} ;
                }
    }
    if (next < n) the graph Gw contains a directed cycle
    else stop ;
}
```

the current object representing a Web page. The cycle detection algorithm is represented as follows.

4.3 The Model Calibration

It is natural and not trivial that the Web is changing dynamically. The Web structure should also cope with the changes with respect to users needs, updating contents, alternating paths, etc. We have to calibrate our model in terms of the changes in detail. When the query terms are altered, measures of the Web Node are also altered. In this case, sensitivity analysis is used to determine whether the entire problem need to be reformulated and recalculated or not. The standard IP problem form separated basic variable between nonbasic variable. After applying a simplex algorithm, above IP problem is reformed.

The criteria of optimality in a simplex algorithm is that an objective function's coefficient of nonbasic variable, i.e., $\bar{c_j} = C_{BV}B^{-1}a_j - c_j$ (a_j is a column vector of N for x_j , c_j is objective function's coefficient value for nonbasic variable x_j) must be non negative. Also, the feasibility condition of the current basis solution is that RHS of the equation (5)-2, i.e., $B^{-1}b$ must be non-negative. When the weight of the Web node is changed, but if this change does not influence the above two conditions (i.e., the optimality and feasibility condition), the current basis is conserved. Further details will be discussed in the following section.

5. Robustness by Sensitivity Analysis

A digraph example consisting of 7 Web pages, i.e., N_0 to Web page N_6 is represented in [Figure 2]. The weight of an edge is assumed to have the average of the content weight

from the two terminal nodes in the edge. After reorganizing the problem as a standard form like (4)-1 and (4)-2, each of the BV (Basic variable), NBV (Nonbasic variable), objective *function* coefficient c_{BV}, c_{NBV}, basis matrix B, the RHS is described in the Appendix where changes to the IP problem induced by Web page modification is represented.

When the Web page is changed, the corresponding objective function's coefficients also are influenced. If it does not break primal feasibility condition, i.e. $B^{-1}b \geq 0$, then the current basis will not change. Consider the case where W_4 value changes the current 9.66 to $9.66+\theta$. Then each of c_{14}, c_{41}, c_{04}, c_{40}, c_{24}, and c_{46} is changed as follows.

c_{14} : (3.45 + 9.66)/2 → (3.45 + 9.66 + θ)/2
c_{41} : (3.45 + 9.66)/2 → (3.45 + 9.66 + θ)/2
c_{04} : (9.49 + 9.66)/2 → (9.49 + 9.66 + θ)/2
c_{40} : (9.49 + 9.66)/2 → (9.49 + 9.66 + θ)/2
c_{24} : (6.21 + 9.66)/2 → (6.21 + 9.66 + θ)/2
c_{46} : (9.66 + 12.42)/2 → (9.66 + θ + 12.42)/2

In this case, the objective *function*'s coefficient that is changed does not affect basis matrix B as well as the objective *function*'s coefficient of the basic variable. So, if all nonbasic variable's $\bar{c_j} = C_{BV}B^{-1}a_j - c_j$ is nonnegative then the current basis remains unchanged. Then the $C_{BV}B^{-1}$ is changed as follows.

〈Table 1〉 Relationship between alteration of Homepage and IP standard form

	Modification in Web page	Modification in IP standard form
Topology is conserved	Alteration of the Web page's *tf-idf*	Alteration of objective coefficient (c_{BV}, c_{NBV})
Topology is modified	Insertion of the new link (previous pages are conserved)	Alteration of matrix **N** (Nonbasis coefficient aij for constraints ($x_{ij} = 0 \quad \forall\ i,j \notin N, \cdots\cdots, 3-(4)$) that represent newly inserted link is changed from 1 to 0)
	Deletion of the link (previous pages are conserved)	Insertion of the constraints (constraint $x_{ij} = 0 \quad \forall\ i,j \notin N, \cdots\cdots, 3-(4)$)
	Insertion of a new Web page	Alteration of objective coefficient (c_{BV}, c_{NBV}), Column insertion of matrix **N**, Insertion of the constraints (1) Below indegree 1 (2) Constraints for cycle resolution, if any (in corresponding case) (3) Constraints for self-cycle resolution (4) Link constraints for a newly inserted Web page

$C_{BV}B^{-1}$ = [9.575 + θ/2, 6.47, 9.575 + θ/2, 10.005, 0, 8.625, 11.04 + θ/2, 0, 7.85, 0, 0, 0, 9.575, 9.49, 0, 0, 7.59, 0, 0, 0, 0, 6.47, 0, 0, 0, 0, 7.85, 0, 0, 0, 8.54, 0, 0, 0, 10.005, 6.555 + θ/2, 7.935, 0, 0, 0, 0, 8.625, 0, 0, 0, 9.315, 11.04, 11.04] * inv(B)[1)]

= [9.575 + θ/2, 6.47, 7.85, 10.005, 9.575 + θ/2, 8.625, 11.04 + θ/2, 0, 0, 0, 0, 0, -8.625 + 9.575, -9.575-θ/2 + 9.49, -6.47, -7.85, -10.005 + 7.59, -9.575-θ/2, -8.625, -11.04, -11.04, -9.575-θ/2 + 6.47, -7.85, -10.005, -8.625, -11.04-θ/2, -9.575 - θ/2 + 7.85, -6.47, -10.005, -8.625, -9.575 - θ/2 + 8.54, -6.47, -7.85, -9.575-θ/2, -11.04-θ/2 + 10.005, -6.47 + 6.555 + θ/2, -7.85 + 7.935, -10.005, -8.625, -6.47, -7.85, -10.005 + 8.625, -9.575 - θ/2, -9.575 - θ/2, -6.47, -7.85 + 9.315, -9.575-θ/2 + 11.04, -8.625 + 11.04]

1) The notation inv() means the inverse matrix operator.

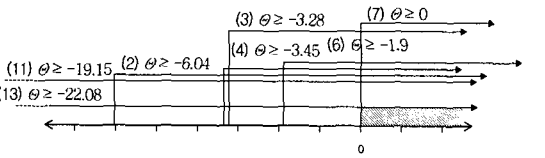
Of course, above $C_{BV}B^{-1}$(except Θ) equals previous $C_{BV}B^{1}$. Now, all nonbasic variable's objective *function*'s coefficient is changed. Considering all of the range of the equations, the results are summarized in [Figure 3]. It can be seen from these results that the current basis is maintained as long as $\Theta \geq 0$. In other words, although N_4's weight increases infinitely, the current basis remains unaffected by such change. Consider the case where node N_4 and N_3 is connected from N_4 to N_3. In matrix N, the coefficient of variable x_{43}, i.e., a_{43}'s value is changed from 1 to 0. In this case, x_{43} is a nonbasic variable, so the nonbasic matrix N's change does not affect the feasibility($\mathbf{B}^{-1}\mathbf{b} \geq 0$) and optimality($\bar{c}_j = C_{BV}B^{-1}a_j - c_j \geq 0$) condition. Hence, the only coefficient that may change from positive to negative, i.e., $\bar{c}_{43} = C_{BV}B^{-1}a_{43} - c_{43}$'s sign confirmation is sufficient for determining basis alteration. The coefficient $\bar{c}_{43} = C_{BV}B^{-1}a_{43} - c_{43}$ is changed as follows.

$$\bar{c}_{x43} = C_{BV}B^{-1}a_{x43} - c_{x43}$$

$$= C_{BV}B^{-1}[0001000000000000000000000000000000000010000000000]^{T} - 0$$

$$= 10.005 - 10.005 - 0 \geq 0$$

The feasibility condition for current solution and the optimality condition is maintained. This also means that the current basis is not changed. In case of a link is deleted, the constraint that forms as $x_{ij} = 0 \ \forall i,j \notin N$ is inserted into the model. If the current solution does satisfy the new constraint, then the current basis is maintained, otherwise, dual simplex algorithm can be used. Considering the case where the link between node W_3 and W_4 are disconnected, then the constraints $x_{34} = 0$ should be included into the previous model. As the current solution satisfies the inserted constraints, the current basis is maintained. The objective *function* coefficient, the element of **N** matrix for the new inserted variable, and all constraint type except the non-negative constraint can be included in the model so that the current solution can not be maintained. So a new constraint can be appended and the new problem should be solved by the same process.

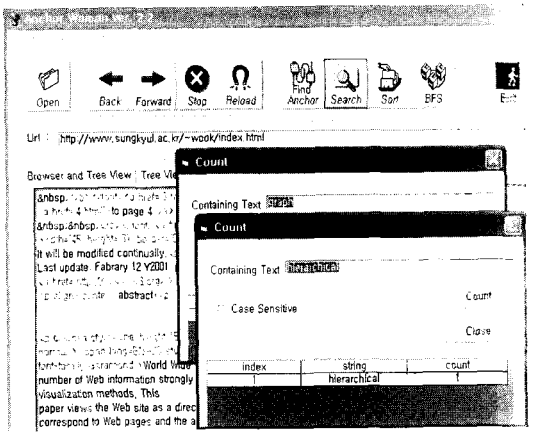


[Figure 7] Determination of range of Θ for which current basis remains optimal

6. Description of System Requirement

6.1 Configurations

A prototype system was developed to provide users with higher-level summaries and with the structure of Web sites. This system provides a site map, a browser, and a node weight tap. The prototype system consists of sub-components as a keyword selection module [Figure 4], a preprocessing module [Figure 5], Node and Edge table generator, and IP solver. The system was implemented with VB 6.0 as the client session and Microsoft ACCESS 2000 as the server database. The system begins with the homepage predetermined by the Web server and uses the interior anchors in the homepage.



[Figure 4] Keyword selection module

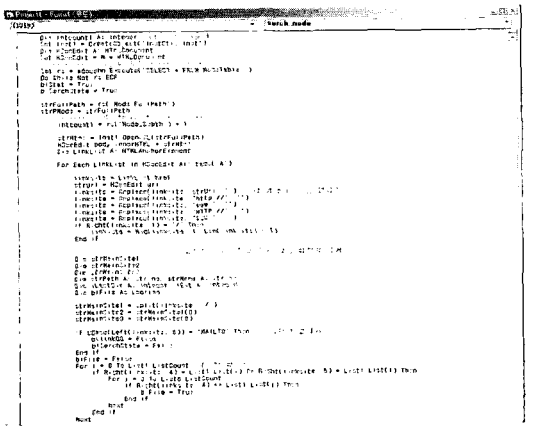
The anchors are classified into two categories, i.e., interior anchors and exterior anchors. If the homepage begins with a frame without anchors, the system extracts the anchors from the frame which includes HTML file. The result anchors with page contents are stored in the database. The anchors can be expanded in the site map of the system and an example is represented as shown on the left-hand side of [Figure 6] and the keyword selection module is in the front pop window.

6.2 System Preparation

The AnchorWoman system parses a given web site through the preprocessing procedure and extracts the corresponding Web information and transfer them to a server database. [Figure 5] represents core modules for the AnchorWoman's preprocessing procedure to search the Web node with which the system produces the subordinate information in the database.

In the server database, the schemata for NodeTable and EdgeTable are created. NodeTable's schema consists of the following attributes : "root page's absolute address", "ID for created nodes(pages)", "root page's filename," and "key index number for created nodes(pages)". The attributes for EdgeTable are "Edge _ID", "Parent Node," and "Child Node" respectively.

The source codes in [Figure 5] represents that HTML Document Object and its methods, i.e., "Alltags("A")" and "OpenURL('URL')". These find the anchor tags(<a>) from the web pages which is parsed by this procedure. Each anchor tag whose target Edge address is parsed to identify internal links [16, 21]. The target link addresses are incorporated into the NodeTable and the EdgeTable having file extensions such as "HTML", "HTM", "PHP" and "ASP" etc. The purpose of this procedure is to identify the normal web page without noise link objects such as "*.mpeg", "*.jpeg", etc. After parsing each page from the root page to the terminal pages, all attributes are stored sequentially in the NodeTable and the EdgeTable.



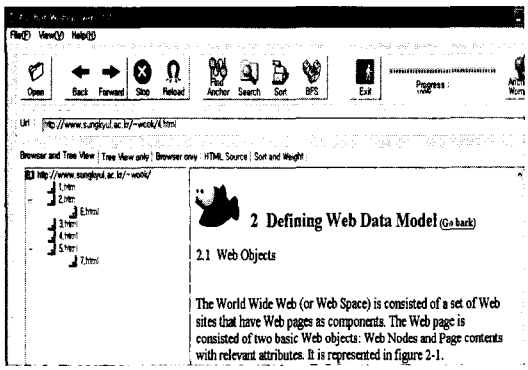
[Figure 5] AnchorWoman's Preprocessing part

6.3 Functionalities for IP solver

We used LINDO 6.1(LINDO API 3.0) as a tool for solving IP model in the AnchorWoman system. The core functions in the AnchorWoman consists of the following three basic

components : IP maker, IP solver, and IP reporter. The IP maker generates an IP problem formulation from which the LINDO's API obtains the data in the previous preprocessing procedure. The IP solver runs the core function of LINDO and generates the optimal solution for the IP problem. IP reporter represents optimal solution and runs the sensitivity analysis.

6.4 The Prototype System



[Figure 6] An Example view of test Web site with the AnchorWoman system

The AnchorWoman prototype system generates a Web site graph as follows. The prototype system is developed to provide users with higher-level summaries and with the structure of Web sites. The system provides a site map, a browser, and a node weight tap. The system begins with a homepage predetermined by the Web server and uses the interior anchors in the homepage and the result anchors with page contents are stored in the database. In the right hand side, a browser shows the Web page where the token is placed in the object Web site. In the [Figure 6], a test Web site is placed and analysed by the AnchorWoman system, and the optimal hierarchical structure in the left hand side of the [Figure 6].

7. Concluding Remarks

We formalize the view on the Web with respect to an optimization approach for generating the Web nodes and Web edges from Web sites. A mathematical model is pertained onto Web sites in terms of integer programming by which the Web graph can be converted to the hierarchical optimal structure. A prototype system called AnchorWoman (ver 2.2) is implemented in terms of a content centric measure derived from tf-idf that represents the keyword semantics of user's intention, and so it is recently widely accepted as one of the most popular information retrieval systems. The scheme from the structure and the corresponding sensitivity analysis yields allowable ranges in maintaining the optimal solution, which prevents the Web structure from totally reconstructing by a minor modification.

Future research issues are as follows. One is to extend our approach to cope with various measures not only on the vector space model but also on the link based approaches as PageRank or HITS. The other is to derive the optimal model from various ranges and domains such as business processes, constrained models, strongly coupled components in the WWW. Implementation challenges need to be worked for an effective and efficient search engine in terms of the optimization approach that includes issues such as Web edge cycles, path generations, semantics on similarity distances, etc.

Acknowledgements

This work was partially supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

References

[1] Brin, S. and Page, L., "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks*, Vol.30, No.1-7 (1998), pp.107-117.

[2] Broder, A., M. Najork and J. Wiener, "Efficient URL Caching for World Wide Web Crawling," *WWW* (2003) pp.679-689.

[3] Chen, M., M. Hearst, J. Hong and J. Lin, "Cha-Cha : A System for Organizing Intranet Search Results," *USENIX on Internet Technologies and Systems*(1999), pp.11-14.

[4] Cooley, R., "The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns," *ACM Internet Technology*, Vol.3, No.2(2003), pp.93-116.

[5] Demaine, E.D. and A. Lopez-Ortiz, "A Linear Lower Bound on Index Size for Text Retrieval," *Journal of Algorithms*, Vol.48, No.1 (2003), pp.2-15.

[6] Garofalakis, J., P. Kappos and D. Mourloukos, "Web Site Optimization Using Page Popularity", *IEEE Internet Computing*, Vol.3, No.4 (1999), pp.22-29.

[7] Glover, E.J., K. Tsioutsiouliklis, S. Lawrence, D.M. Pennock and G.W. Flake, "Using web Structure for Classifying and Describing Web Pages," *WWW* (2002), pp.562-569.

[8] Gurrin, C. and A.F. Smeaton, "Replicating Web Structure in Small-Scale Test Collections," *Information Retrieval*, Vol.7, No.3-4 (2004), pp.239-263.

[9] Gabriel Nivasch, "Cycle detection using a stack," *Information Processing Letters*, Vol. 90, No.3(2004), pp.135-140.

[10] Henzinger, M.R., A. Heydon, M. Mitzenmacher and M. Najork, "On Near-uniform URL Sampling", *Computer Networks*, Vol.33, No.1(2000), pp.295-308.

[11] Hou, J. and Y. Zhang, "Effective Finding Relevant Web Pages from Linkage Information", *IEEE TKDE*, Vol.15, No.4(2003), pp. 940-951.

[12] Kumar, R., P. Raghavan, S. Rajagopalan and A. Tomkins, "Trawling the Web for Emerging Cyber-Communities", *WWW* (1999), pp. 403-415.

[13] Gurrin, C. and A.F. Smeaton, Replicating Web Structure in Small-Scale Test Collections, Information Retrieval, Vol.7, No.3 (2004), pp.239-263.

[14] Mendelzon, A.O. and T. Milo, "Formal Model of Web Queries," *ACM PODS* (1997), pp.134-143.

[15] Demaine, E., A. Lopez-Ortiz, "A Linear Lower Bound on Index Size for Text Retrieval," *Journal of Algorithms*, Vol.48, No.1(2003), pp.2-15.

[16] Najork, M. and J. Wiener, "Breadth-first crawling yields high-quality pages," *WWW* (2001), pp.114-118.

[17] Pandurangan, G., P. Raghavan and E. Upfal, "Using PageRank to Characterize Web Structure", *COCOON* (2002), pp.330-339.

[18] Glover, E.J., K. Tsioutsiouliklis, S. Lawrence, D.M. Pennock and G. Flake, "Using Web Structure for Classifying and Describing Web Pages", *WWW* (2002) pp.562-569.

[19] Subramani, K. and L. Kovalchick, "Contraction versus Relaxation : A Comparison of Two Approaches for the Negative Cost Cycle Detection Problem," *Computational Science* (2003), pp.377-387.

[20] Thom, L.H. and C. Iochpe, "Integrating a Pattern Catalogue in a Business Process Model," *In Proc. ICEIS* (2004), pp.651-654.

[21] Wookey, L. and J. Geller, "Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers," *Journal of Research and Practice in Information Technology*, Vol .36, No.1(2004), pp.71-82.

[22] Zwol, R. and P. Apers, "The webspace method : On the Integration of Database Technology with Multimedia Retrieval," *ICIKM* (2000), pp.438-445.

Appendix

〈Table 2〉 Standard form representation of example problem

	Basis	Nonbasis
Variable	$x_{BV}=[x_{40}, x_{01}, x_{04}, x_{63}, s_8, x_{35}, x_{46}, s_7, x_{02}, s_9, s_{10}, s_{11}, x_{05}, x_{00}, x_{11}, x_{22}, x_{33}, x_{44}, x_{55}, x_{66}, x_{06}, x_{10}, x_{12}, x_{13}, x_{15}, x_{16}, x_{20}, x_{21}, x_{23}, x_{25}, x_{30}, x_{31}, x_{32}, x_{34}, x_{36}, x_{41}, x_{42}, x_{43}, x_{45}, x_{51}, x_{52}, x_{53}, x_{54}, x_{60}, x_{61}, x_{62}, x_{64}, x_{65}]^T$	$x_{NBV}=[x_{03}, x_{14}, x_{24}, x_{26}, x_{43}, x_{50}, x_{56}, s_1, s_2, s_3, s_4, s_5, s_6, s_{12}]^T$
Objective function coefficients	c_{BV}=[9.575, 6.47, 9.575, 10.005, 0, 8.625, 11.04, 0, 7.85, 0, 0, 0, 9.575, 9.49, 0, 0, 7.59, 0, 0, 0, 0, 6.47, 0, 0, 0, 0, 7.85, 0, 0, 0, 8.54, 0, 0, 0, 10.005, 6.555, 7.935, 0, 0, 0, 0, 8.625, 0, 0, 0, 9.315, 11.04, 11.04]	c_{NBV}=[8.54, 6.555, 7.935, 9.315, 0, 9.575, 11.04, 0, 0, 0, 0, 0, 0, 0]
Constraint Matrix2)	B=[100000000000001000000001000010001000000000010000 010000000000000100000000000100010001000100001000 000000001000000100000010000000001000100010000100 000100000000000001000000100001000000010001000000 001000000000000001000000000000000100000000100010 000001000000100000100000100001000000000100000001 000000100000000000001100001000000000100000000000 110000010000000000000000000000000000000000000000 100010001000000000000000000000000000000000000000 000001000100000000000000000000000000000000000000 000101000010000000000000000000000000000000000000 101000000001000000000000000000000000000000000000 000000000000100000000000000000000000000000000000 000000000000010000000000000000000000000000000000 000000000000001000000000000000000000000000000000 000000000000000100000000000000000000000000000000 000000000000000010000000000000000000000000000000 000000000000000001000000000000000000000000000000 000000000000000000100000000000000000000000000000 000000000000000000010000000000000000000000000000 000000000000000000001000000000000000000000000000 000000000000000000000100000000000000000000000000 000000000000000000000010000000000000000000000000 000000000000000000000001000000000000000000000000 000000000000000000000000100000000000000000000000 000000000000000000000000010000000000000000000000 000000000000000000000000001000000000000000000000 000000000000000000000000000100000000000000000000 000000000000000000000000000010000000000000000000 000000000000000000000000000001000000000000000000 000000000000000000000000000000100000000000000000 000000000000000000000000000000010000000000000000 000000000000000000000000000000001000000000000000 000000000000000000000000000000000100000000000000 000000000000000000000000000000000010000000000000 000000000000000000000000000000000001000000000000 000000000000000000000000000000000000100000000000 000000000000000000000000000000000000010000000000 000000000000000000000000000000000000001000000000 000000000000000000000000000000000000000100000000 000000000000000000000000000000000000000010000000 000000000000000000000000000000000000000001000000 000000000000000000000000000000000000000000100000 000000000000000000000000000000000000000000010000 000000000000000000000000000000000000000000001000 000000000000000000000000000000000000000000000100 000000000000000000000000000000000000000000000010 000000000000000000000000000000000000000000000001]	N=[0000010000000 0000000100000 0000000010000 1000100001000 0110000000100 0000000000010 0001001000001 0100000000000 0010000000000 1000010000000 0000001000000 0000000000000 0000010000000 0000000000000 0001000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000]
RHS	b=[0 1 1 1 1 1 1 2 2 2 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]^{Ts}	

2) B=[a b; c d] means $B=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$.S