STRUCTURING THE WEB TO COPE WITH DYNAMIC CHANGES

Wookey Lee[1] and Jinho Kim[2]
[1]Dept. of Computer Science, Sungkyul University, 147-2 Anyang 8 Dong, Anyang, Korea;
[2]Dept. of Computer Science, Kangwon National University, 192-1 Hyoja Dong 2, Chunchon, Kangwon, Korea

Abstract: The web structure yields significant insights into web algorithms for searching, structuring, discovering, mining, and revealing web information. We formalize our view of the web structure in terms of the integer linear programming that converts the web directed graph to the optimal hierarchical structure. The model represents a high level structure regardless of various measures such as the *cosine* similarity and *tf-idf* measure of the vector space model as well as the *PageRank* of Google. Another advantage for our approach is that the corresponding sensitivity analysis yields an allowable range for the optimal structure so that the model can be estimated even though dynamic changes take place in the web pages, links, and structures.

Key words: web structuring, linear programming, VSM, tf-idf, *PageRank*

1. INTRODUCTION

The WWW can be viewed as a hierarchy of web objects. The WWW can be seen as a set of web sites, and a web site can be seen as a digraph with web nodes and arcs, where the web nodes correspond to HTML files having page contents and the arcs correspond to hypertext links interconnected with the web pages. The web-as-a-graph approach can be a starting point to generate a structure that can be used for web site designers, search engines, web crawling robots, and web marketers and analysts.

One of the typical examples of web site structuring is tree modeling, such as breadth first search (BFS), depth first search (DFS), web catalogues or site maps [1, 5, 15]. Complex and static web abstractions, however, do little to help a web designer or a web crawler that wants to model a web site, and also often poses navigational hazards. The structural abstraction is useful in organizing information and reducing the number of alternatives that must be considered at any one time [10].

The BFS, including *backlink* count method or *RageRank* [5, 7, 9], has some advantages so that an 'important' web page can easily be accessed by simply clicking relatively fewer steps from its default page. It is easy to reduce a graph to a tree form and the depths for accessing each web page. Thus, the BFS can statistically minimize the total access time, but structurally it is extremely flat because all pages may stick to the root node.

On the other hand, the DFS is an easy method to adopt, and, from a cognitive science point of view, the search methods are similar to the behaviors of human snoopers. But, the DFS is inappropriate for structuring the web because a long series of web pages means as many clicks as there are pages which, in turn, entails massive time consumption to access each page.

A topological ordering algorithm [2, 14] converted a web site structure to an unbiased hierarchical form; this minimizes the average distance from the root node to a specific web page. They also considered the semantic relevance between the web nodes. The weakness is that when there are minor changes in the web page's weight or in the link structure, 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 under average 20 a page, and that this number will grow logarithmically with the size of the web [2, 25].

We formalize our view of the web structure in terms of the integer programming that converts the web directed graph to the optimal hierarchical structure. The structure of a web site can prevent the search robots or crawling agents from confusing in the midst of huge web pages in the web sites. Therefore, in this paper, we introduce an attribute called, "the weight," to evaluate the significance of web pages.

This paper is organized as follows: In Section 2, we present the data model of web sites, the web schema, and conventional approaches. In Section 3, we discuss several weight measures endowed on web nodes. In Section 4, we will discuss on the integer linear programming model of the web structure. In Section 5, we will present an example of our model and present the robustness of our model with experimental results. Finally, we conclude the paper.

WEB SCHEMA AND SIMILARITY MEASURES

1.1 The Web Schema

A web site can be defined as a set of web nodes $N_w = \{N_1, ..., N_n\}$, a directed graph $G_w = (N_w, E_w)$, an arc function $x_{ij} : N^k \rightarrow \{0, 1\}$, $\forall (i, j) \in E_w$ consisting of a finite web node set N_w, a finite web arc set E_w of ordered pairs of web nodes, and the web arc elements (i, j) respectively, where $i, j \in \{0, 1, 2, 3, ... , n\text{-}1\}$, and $n = |N_w|$ the cardinality of web pages. There is a mapping system for the nodes corresponding to web pages and the arcs to Uniform Resource Identifiers [4, 5]. The web node (N_W) can be defined as follows:

$$N_w = [N_i, \{(i, j), \forall i\}, w_i] \qquad (2\text{-}1)$$

Where the N_i represents a web node corresponding to an HTML file (we set a node identifier as i). Where the homepage is defined as a default page (index.html) predetermined by the web server. The $\{(i, j), \forall i, j \in E_w\}$ is the set of web arcs having hypertext links to which the web page indicates.

There are three approaches to investigate the web digraph domain as (1) the whole web [1, 2, 12], (2) a set of strongly coupled components [9], (3) set of a web site [14, 17]. The first one is utilized to measure the whole size or growing ratio, but it is not appropriate to derive a web structure. The second focuses a mathematical model, and the third is inclined to practical side. So we adopt the third for implementation's sake, where the homepage is defined as a default page is predetermined by the Web server and the other pages are interconnected each other.

1.2 How to Measure the Web nodes?

In order to generate the web structure, we introduce the representative weight measures such as *cosine* measure and *tf-idf* measure from Vector Space Model (VSM), and *PageRank.* For the first time, most web ranking algorithms utilize a similarity measure in terms of VSM, which has been extensively studied in the information retrieval community [6, 8, 13]. For the VSM, to compute the similarities among a set of web pages, each page can be viewed as an n dimensional vector $<w_1,...,w_m>^\mathrm{T}$. The *cosine* similarity [13] between a query, Q, and a web page, $N_{i,}$ can then be defined as the inner product of the Q and n vectors.

Another common way of computing a web page's weight, W, is the *tf-idf*, which is obtained as an normalized vector, $W' = < w'_1,...,w'_m>^\mathrm{T}$, where each w'_i is the product of a term frequency (*tf*) factor and an inverse document frequency (*idf*) factor. The *tf* factor is proportional to the frequency of the i^{th} word within a web page. The *idf* factor is a factor divided by the number of times that 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 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 from which the weight specified as the inner product of the two. There are other measures from VSM in terms of local weights such as the Binary model [13], Logarithmic [14] and global weights such as the entropy model [8], GfIdf [13], and Probabilistic Inverse [18].

There are two major link-based search algorithms, HITS and *PageRank*. The basic idea of the HITS algorithm is to identify a small sub-graph of the web and apply link analysis on this sub-graph to locate the authorities and hubs for the given query. The sub-graph that is chosen depends on the user query. The selections of a small sub-graph (typically a few thousand pages), not only focus the link analysis on the most relevant part of the web, but also reduce the amount of work for the next phase. The main weaknesses of HITS are known to non-uniqueness and nil-weighting. Haveliwala [12] suggested a domain based *PageRank* algorithm, but its limitation depends on the usefulness of the ontological state of the algorithm and the particular thesaurus that the system tries to include in discriminating semantics among web documents.

One of the representative arc oriented approaches is the *PageRank* algorithm [7, 9, 12]. *PageRank* has the arc based measure of a page that is similar to its indegree, which is a possible measure of the importance of a page. The *PageRank* of a page is high if many pages with a high *PageRank* contain links to it, and a page containing few outgoing links contributes more weight to the pages it links to than would a page containing many outgoing links. It is a static measure, so that it is modeled to rank pages in the absence of any queries. So the *PageRank* computes the global worth of each page. The algorithm, however, has a weakness called Google bombing that can be attacked by creating a large number of bogus web pages, all pointing to a single target page. Since many search engines take into account the number of incoming links in ranking pages, the rank of the target page is likely to increase, and appear earlier in query result sets. So, the page can achieve 'higher-than-deserved' rankings in Google [4]. The weights in this paper are assumed to accurately indicate the importance of the web page [19, 20, 21, 22]. We generalized the formula and weight from the previous work [17].

2. SIMILARITY MEASURES

The similarity measure in this paper is the weight of web node. We introduce the *cosine* measure, *tf-idf* measure, and *PageRank* measure as the weight measure which can be used to determine the topological ordering of web sites. The prototype system has been experimentally tested to search for the structure of the test web site. The link structure of the site is shown in reference Fig. 3.1 and 3.2. The circle in the figures represents a web node and the arrow represents a hyperlink or a web arc.

First of all, the *cosine* measure from the VSM is introduced [13] as following equation (3.1) by which the measure is exploited in the experiment

$$weight(\mathrm{W}_j) = \frac{a_j^{\mathrm{T}} q}{\| a_j \|_2 \| q \|_2} = \frac{\sum_{i=1}^{m} a_{ij} q_i}{\sqrt{\sum_{i=1}^{m} a_{ij}^2}\sqrt{\sum_{i=1}^{m} q_i^2}} \tag{3.1}$$

In the *tf-idf* measure, the *tf* factor itself is sometimes normalized by dividing it by the frequency of the most-frequent non-stop term in the document as, $tf_{\mathrm{norm}} = tf/tf_{\mathrm{max}}$. The *idf* factor is typically computed by $[df(w_i)/N]^{-1}$, and most often the $log_2[N/ df(w_i)]$ is used, where, *N*, is the total number of documents and, $df(w_i)$, is the number of documents containing the i^{th} word. Refer to equation (3.2) [8].

$$\begin{aligned} weight(W_j) = Q * W_j &= [q_i] * \left[(tf_i \cdot \log_2 [N / df(w_i)])\right]^T \\ &= \sum_i q_i \cdot (tf_i \cdot \log_2 [N / df(w_i)]) \end{aligned} \tag{3.2}$$

The *PageRank* measure is that, if source page, *i*, has a link to target page, *j*, then the author of source page, *i*, is implicitly conferring some importance to page, *j*. Let N_j be the out-degree of page, *i*, and let *Rank*(*p*) represents the importance of page, *p*. Then, the link (*i*, *j*) confers a certain number of units of rank to, *j*. This simple idea leads to the following iterative fix-point computation that yields the rank vector over all of the pages on the web. If, *n*, is the number of pages, assign all pages the initial value 1/n. Let, B_j represent the set of pages pointing to *j*. Links between web pages propagate the ranks [9]. We continue the iterations until the rank is stabilized to within some defined threshold. The final rank vector contains the *PageRank* vector over the web. This vector is computed only once after each crawl of the web; the values can then be used to influence the ranking of search results [6]. Guaranteeing the rank vector to converge, *PageRank* algorithm uses the following equation with a damping factor (*d*):

$$\forall i, Rank^{(k+1)}(i) = (1-d)E + d\left(\sum_{i \in B_j} Rank^{(k)}(j) / N_j\right) \quad where,\ E = \left[\frac{1}{n}\right]_{n \times 1} \tag{3.3}$$

In Google, we usually set the value of the damping factor to 0.85 [7, 8] so that we can see that the *PageRank* vector converges either slowly or quickly in terms of the magnitude of the damping factor. For example, by (3.3) we can get the weights in Table 3.1 and derived weights as Fig. 3.1. The node weights by *PageRank* are: <0.865, 0.479, 0.50, 2.25, 0.64, 1.00, 1.25>. The link weights are generated by simple Euclidean distance between the web nodes as: <x02, x01, x14, x41, x04, x24, x25, x45, x03, x53, x06, x36, x65> = <0.664, 0.80, 0.777, 0.777, 0.984, 0.640, 0.846, 1.166, 1.220, 1.402, 1.161, 1.374, 1.343>.

Table 3.1 *PageRank* values for Fig. 3.1

N0	N1	N2	N3	N4	N5	N6
1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.858	0.603	0.745	1.878	1.028	1.141	0.745
0.757	0.587	0.552	2.275	0.763	0.969	1.094
0.831	0.495	0.528	2.226	0.684	0.987	1.245
0.873	0.485	0.501	2.226	0.651	1.023	1.237
0.860	0.483	0.504	2.256	0.646	1.002	1.244
0.862	0.479	0.501	2.250	0.644	1.005	1.255
0.866	0.479	0.500	2.249	0.642	1.008	1.253
0.864	0.479	0.500	2.252	0.642	1.006	1.253
0.864	0.479	0.500	2.251	0.642	1.006	1.254
0.865	0.479	0.500	2.251	0.642	1.006	1.253

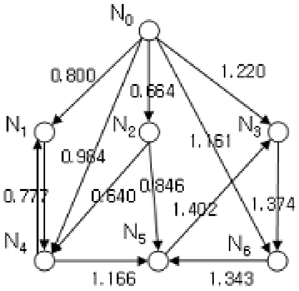


Fig. 3.1 Topology for *PageRank* on the example site

3. MATHEMATICAL MODELING

The model can be applied to transform a digraph into a tree. First, except for the root node, the tree node's indegree should be 1. Second, All cycles which may exist in a graph should be removed during the transformation phase. Cycle detection is implemented by DFS using Stack based algorithm

[25] in which results the constraints (4.4) to (4.6). Third, a self-cycle within a graph node should be removed. And last, duplicate paths between two adjacent nodes in a graph should be removed. So, considering the above constraints, IP formulation is as follows:

$$\max \sum_{\substack{i,j \in N, j \ge 1 \\ (i,j) \in A}} w_{ij} x_{ij} \tag{4.1}$$

$$s.t \sum_{\substack{i \in N \\ (i,0) \in A}} x_{ij} = 0 \tag{4.2}$$

$$\sum_{\substack{i \in N \\ (i,j) \in A}} x_{ij} = 1 \ \text{ for all } j \ne 0 \tag{4.3}$$

$$x_{ii} = 0 \quad \text{for all } i \tag{4.4}$$

$$x_{ij} + x_{ji} \le 1 \quad \forall i, j \tag{4.5}$$

$$x_{ij_1} + \sum_{k=1}^{m-1} x_{j_k j_{k+1}} + x_{j_m i} \le m \quad \text{for } 2 \le m \le |N| - 1 \tag{4.6}$$

$$x_{ij} = 0 \quad \forall i, j \notin N \tag{4.7}$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i, j \tag{4.8}$$

The variable x_{ij} is 1 if there exists a web arc from node *i* to node *j*, zero otherwise. Set ***A*** represents a set of all existing links between any of two Web pages in a Web site, where notation (*i*, *j*) is used to represent a link from node *i* to node *j*. The weight parameter w_{ij} represents the average distance from node *i* to node *j*. There are several alternatives for deriving the weight of a node and to generate a distance to the link, including the number of inward or outward links [24]. In this paper, we use weight measures as tf-idf, Cosine, and *PageRank* to each node, and generate a Euclidean distance w_{ij} from node *i* to node *j*. The objective function (4.1) which is to be maximized denotes the sum of weights for links. The constraint (4.2) ensures that any of the links toward the root page should not be included for making feasible spanning tree, while constraint (4.3) ensures that each of all web pages should have only one web page that is directly linked to a parent node except the root node. The constraint (4.4) removes a self-cycle. The constraints (4.5) and (4.6) are to remove a cycle with 2 steps and a cycle by more than 3 steps, respectively. Constraint (4.6) means that for all cycles starting from any of Web page *i* in a given Web site, where a cycle starting from Web page *i* and returning to the same page through *m* intermediate

Web pages, can be represented as an ordered list of Web pages (i, j_1, j_2,..., j_k, j_{k+1},..., j_{m-1}, j_m, i). The constraint (4.7) restricts the domain within a web site. The constraint (4.8) representing the problem should be binary integers.

4. OPTIMAL STRUCTURE AND CHANGE ANALYSIS

When the query terms are altered, measurements of the web Node are also altered. In this case, sensitivity analysis is used to determine whether the entire problem needs to be reformulated and recalculated or not. The standard IP problem forms separated basic variables between nonbasic variables. After applying a simplex algorithm, the above IP problem is reformed.

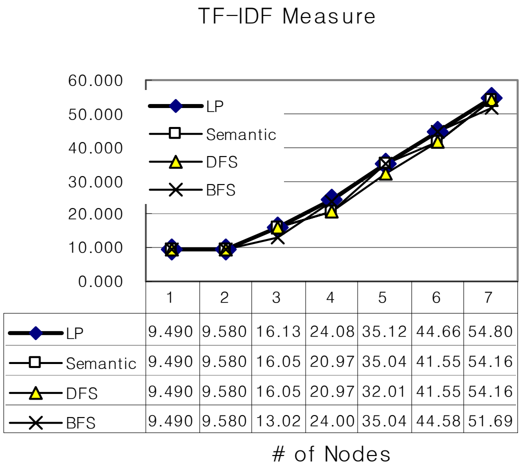
The criteria of optimality in a simplex algorithm is that an objective function's coefficient of a non-basic variable, i.e., $\overline{c}_j = C_{BV}B^{-1}a_j - c_j$ (a_j is a column vector of N for x_j , c_j is objective function's coefficient value for a nonbasic variable x_j) must be non-negative.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

Fig. 3.1 represents a digraph consisting of 7 web pages, i.e., N_0 to web page N_6. The weight of an arc is the mean average weight value of two terminal nodes of the arc. After reorganizing the IP problem, each of the BV (Basic variable), NBV (Nonbasic variable), objective *function* coefficient c_{BV}, c_{NBV}, basis matrix B, RHS (Right Hand Side) can be described.

If the web page is changed, then the corresponding objective function's coefficients are also influenced. If it does not break the primal feasibility condition, i.e. $B^{-1}b \geq 0$, then the current basis will not change. In this case, the current solution's feasibility condition and optimality condition is maintained. This also means that the current basis is not changed. In case a link is deleted, the constraint can be inserted into the formulation. If the current solution does satisfy the new constraint, the current basis is maintained, otherwise, a dual simplex algorithm can be used. Consider the case where two links between nodes, W_1 and $W_{4,}$ are disconnected. Then, the two constraints(x_{14}=0 and x_{41}=0) should be included into the previous IP formulation. As the current solution satisfies the inserted constraints, the current basis is maintained. In this case, the objective *function* coefficient, the element of **N** matrix for the new inserted variable, and all constraint types, except non-negative constraints, can be included in LP formulation. In this case, the current basis is not conserved.

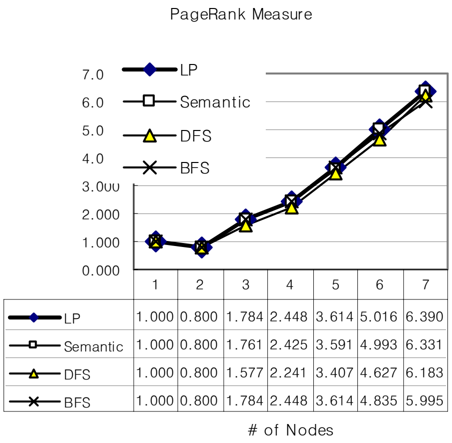
5. EXPERIMENTS

By the example site in section 2, the number of nodes is increased from 1 to 7. The node weights are derived from the *cosine* of *VSM*, *tf-idf*, and *PageRank*. A total of four structuring methods are applied such as DFS, BFS, Semantic, and LP.



	1	2	3	4	5	6	7
LP	9.490	9.580	16.13	24.08	35.12	44.66	54.80
Semantic	9.490	9.580	16.05	20.97	35.04	41.55	54.16
DFS	9.490	9.580	16.05	20.97	32.01	41.55	54.16
BFS	9.490	9.580	13.02	24.00	35.04	44.58	51.69

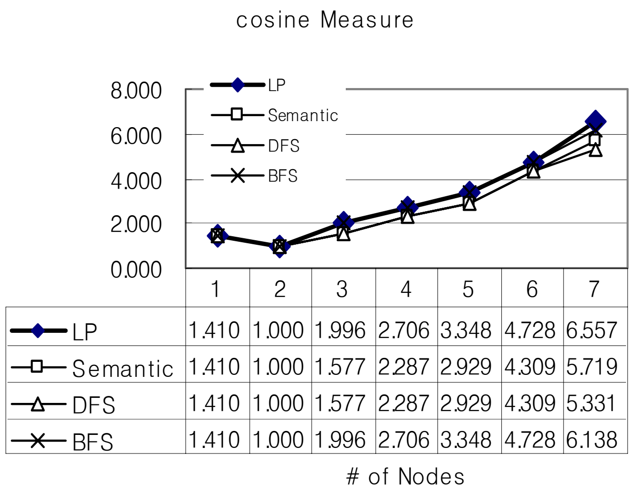
Fig. 6.1 *tf-idf* weight measure traverses w.r.t the number of node



	1	2	3	4	5	6	7
LP	1.000	0.800	1.784	2.448	3.614	5.016	6.390
Semantic	1.000	0.800	1.761	2.425	3.591	4.993	6.331
DFS	1.000	0.800	1.577	2.241	3.407	4.627	6.183
BFS	1.000	0.800	1.784	2.448	3.614	4.835	5.995

Fig. 6.2 *PageRank* weight measure traverses w.r.t. the number of nodes

We conclude that the LP solution produces the optimal solution with optimal structure. In other words, our approach generates the optimal hierarchical structure among many structuring methods. Our method shows its openness of the mechanism without respect to the weight measure such as *cosine* measure, *if-idf* measure, and *PageRank* measure.



	1	2	3	4	5	6	7
LP	1.410	1.000	1.996	2.706	3.348	4.728	6.557
Semantic	1.410	1.000	1.577	2.287	2.929	4.309	5.719
DFS	1.410	1.000	1.577	2.287	2.929	4.309	5.331
BFS	1.410	1.000	1.996	2.706	3.348	4.728	6.138

Fig. 6.3 Cosine weight measure traverses w.r.t the number of nodes

6. CONCLUSIONS

The goal of this paper has been to generate the optimal hierarchical structure of a web site from directed graphs. With respect to the weight measure, the optimal solution can be derived: from a user's point of view, the optimum approach will be consonant with the user's preferences in terms of query with *tf-idf* or *cosine* measures, on one hand; or, from a web robot's point of view, the optimum will be represented by the search robot's point of view. This model guarantees that the optimal hierarchical structure will be generated with respect to the measurement and the optimization model. A mathematical model is pertained onto a web site in terms of the integer linear programming by which the web digraph can be converted to the hierarchical structure. The optimal solution from the hierarchical structure and the corresponding sensitivity analysis yield a robustness of the model in maintaining the solution in a dynamic manner.

The model can secure the optimum under any weight measure conditions, and by using the LP model, a user is always able to get the optimal result even though any actions, such as Adding, Deleting, Inserting, etc., are undertaken within a web site. Under the circumstances of frequent web contents change, the LP model can be a very good tool for developing an advanced search engine schematic. The LP model also shows its openness: Using significant similarity measures (such as *tf-idf*, *cosine* measure, and *PageRank*) does not affect the optimal model, the corresponding solution, and sensitivity analysis, which implies that the model can easily be adapted by and integrated into the current Web search robots as well as the web site managers having various similarity measures.

ACKNOWLEDGEMENT

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

REFERENCES

1. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Crawling the Web for cyber communities in the Web, In Proc. *8th WWW* (1999) 403–415
2. Barabasi A., Albert R., and Jeong H.: Scale-free Characteristics of Random Networks: the Topology of the World-Wide Web. *Physica A* 281 (2000) 69-77.
3. Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. *AAAI* (2004) 391-398.
4. Gyöngyi, Z., Garcia-Molina, H., and Pedersen, J.: Combating Web Spam with TrustRank. *VLDB* (2004) 576-587.
5. Henzinger, M. R., Heydon, A., Mitzenmacher, M. and Najork, M.: On near-uniform URL sampling, *Computer Networks*, 33(1) (2000) 295-308.
6. Demaine, E., Lopez-Ortiz, A.: A Linear Lower Bound on Index Size for Text Retrieval, *Journal of Algorithms*, 48(1) (2003) 2-15.
7. Thom, L. H., and Iochpe, C.: Integrating a Pattern Catalogue in a Business Process Model, *ICEIS* 3 (2004) 651-654.
8. Glover, E. J., Tsioutsiouliklis, C., Lawrence, S., Pennock, D., and Flake, G.: Using Web Structure for Classifying and Describing Web Pages. *WWW* (2002) 562-569.
9. Pandurangan, G., Raghavan, P. and Upfal, E.: Using *PageRank* to Characterize Web Structure. *COCOON* (2002) 330-339.

10. Gabriel Nivasch, "Cycle detection using a stack," *Information Processing Letters*, 90(3) (2004) pp.135-140.
11. Cooley, R.: The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns. *ACM Internet Technology*, 3(2) (2003) 93-116.
12. Mendelzon, A. O. and Milo, T.: Formal Model of Web Queries, *ACM PODS* (1997) 134-143.
13. Berry, M. and Browne, M.: *Understanding Search Engines: Mathematical Modeling and Text Retrieval*, Siam (1999).
14. Wookey Lee and Geller J.: Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers. *Journal of Research and Practice in Information Technology*, 36 (1) (2004) 71-82.
15. Gurrin, C., and Smeaton, A. F.: Replicating Web Structure in Small-Scale Test Collections, *Information Retrieval*, 7(3) (2004) 239-263.
16. Zwol, R. and Apers P.: The Webspace Method: On the Integration of Database Technology with Multimedia Retrieval. *CIKM* (2000) 438-445.
17. Wookey Lee, Seung Kim, Suk-Ho Kang, "Dynamic Hierarchical Website Structuring Using Linear Programming," *LNCS*, Vol. 3182, (2004) 328-337.
18. Glover, E. J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D. M., and Flake G.: Using Web Structure for Classifying and Describing Web Pages. *WWW* (2002) 562-569.
19. Cho, J., and Garcia-Molina, H.: Effective Page Refresh Policies for Web Crawlers. *ACM TODS* 28(4) (2003) 390-426.
20. Chen, M., Hearst, M., Hong, J. and Lin, J.: Cha-Cha: A system for Organizing Intranet Search Results. *USENIX ITS* (1999) 11-14.
21. Subramani, K. and Kovalchick L.: Contraction versus Relaxation: A Comparison of Two Approaches for the Negative Cost Cycle Detection Problem. *Computational Science* (2003) 377-387.
22. Garofalakis, M., Kappos, P. and Mourloukos, D.: Web Site Optimization Using Page Popularity, *IEEE Internet Computing*, 3(4) (1999) 22-29.
23. Hou, J., Zhang, Y.: Effective Finding Relevant Web Pages from Linkage Information. *IEEE TKDE* 15(4) (2003) 940-951.
24. Takeuchi, F.: Nonregular Triangulations, View Graphs of Triangulations, and Linear Programming Duality. *Discrete and Computational Geometry* (2000) 330-338.
25. Shmueli, O.: Dynamic Cycle Detection. *Information Processing Letters*. 17(4) (1983) 185-188.