Hierarchical Web Structuring Using Integer Programming

Wookey Lee[*], Seung Kim[**], Hando Kim[***], Suk-Ho Kang[**]

[*]Dept. of Computer Engineering, Sungkyul Univ., Anyang-8-dong, Manan-gu, Anyang, Korea
wook@sungkyul.edu

[**]Dept. of Industrial Engineering, Seoul National Univ., San 56-1, Sillim-dong, Gwanak-gu, Seoul, Korea
{seung2@netopia, shkang@cybernet}.snu.ac.kr

[***]KEDCOM Co. Ltd., #1122-5, SinGil-Dong, Danwon-gu, Ansan, Korea
hando@kedcom.com

World Wide Web is nearly ubiquitous and the tremendous growing number of Web information strongly requires a structuring framework by which an overview visualization of Web sites has provided as a visual surrogate for the users. We have a viewpoint that the Web site is a directed graph with nodes and arcs where the nodes correspond to Web pages and the arcs correspond to hypertext links between the Web pages. In dealing with the WWW, the goal in this paper is not to derive a naïve shortest path or a fast access method, but to generate an optimal structure based on the context centric weight. We modeled a Web site formally so that a integer programming model can be formulated. Even if changes such as modification of the query terms, the optimized Web site structure can be maintained in terms of sensitivity.

1 Introduction

The World Wide Web is now almost ubiquitous. Web browsers allow users to access on the order of 8 billion Web documents [1]. Because of a vast amount of Web information (pages), as a browser session progresses, the users sometimes feel "being lost in hyperspace" [2, 3]. So, users of Web information, accessible over the global Internet, require assistance by appropriate visualization methods.

The WWW can be viewed as a digraph consisting a bag of Web sites. The Web site is a directed graph 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. A Web site can be viewed as a specific directed graph that consists of an initial node (called homepage) and other nodes connected to it. There is a natural mapping approach of a Web onto a directed graph where the nodes correspond to Web pages and the arcs to URIs (Uniform Resource Identifiers) [4, 5]. A hierarchical structuring such as Web catalogues or super books [6, 7, 8] is one of the typical examples of web site structuring. But complex and static Web abstractions do little to help a Web designer who wants to get modeled within a Web site and often cause navigation problems of their own. The problem of finding a tree structure of a Web site from a directed graph is exponential or NP-hard [9]. What is needed is a virtual Web site structure based on the current query or interest of a user.

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 treat keyword-based weight measure endowed to Web node. We developed a integer programming model and treat sensitivity analysis of proposed IP model in section 4. Then, the example of our model and the comparison of IP method between some relative methodologies are presented in section 5. Finally, we conclude the paper.

2 The Web Data Model

2.1 Web Nodes

We assume that the data model of the World Wide Web consists of a hierarchy of Web objects. The WWW is viewed as a bag of Web sites. The Web schema contains the Meta information that represents a bag of Web pages in a Web site. The Web site is a directed graph 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 node (W_i) is defined as follows:

$$W_i = [\textit{Webpage-Id}, \{\text{URI}\}, \textit{weight}] \qquad (1)$$

Where the W_i represents a node corresponding to an HTML file (without loss of generality we set a node identifier i). Where the homepage is defined as a default page (for example, index.html or default.asp or index.php3, etc.) predetermined by the Web server. The {URI} is the bag of Web nodes having the hypertext links within which the Web page indicates. The *weight* represents the values specified by the measure of keywords (it 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, and multimedia files etc. In this paper, for convenience's sake, the Web page weight is generated by the method described in the following section 3.

2.2 Web Arcs

A Web site can be viewed as a directed graph that consists of an initial node (called the homepage) and the other nodes inter-connected among them. Complex Web representations do little to help the user orientation within the site and usually tolerate navigation problems themselves. A hierarchical abstraction is useful in organizing information and reducing the number of alternatives that must be considered at any one-time [10]. If the Web site can be represented as a hierarchical structure, those problems such as the multiple paths, the recursive cycle, the multi-path cycle, and multiple parents would be resolved. The problem is treated more specifically in [11, 12].

In this paper, the URI's are specified two types: interior arcs and exterior arcs. The interior arcs are the URI's that indicate the HTML files somewhere within the Web site, but the exterior arcs out of the Web site. We are interested in the interior arcs only, for the structure of a Web site is generated in this paper. After preprocessing the URI's of a Web site, then standard (full length) IP addresses of every Web page are derived. The exterior arcs are, however, discarded in the Preprocessing phase, for they have a different server IP address, i.e., a different site.

It can be noted that in some references [12, 13] the interior arcs are additionally classified two types such as interior and local. But it is needless to differentiate the internal arc in more detail. Because once a Web page is transferred from the Web server, there is no need to access the same Web page physically again. Actually in some web sites, there is only a Frame in the default page (ex, index.html). In that case, we give the URI's of the Web pages that the default frame includes.

2.3 Conventional Approaches

Depth first approach (DFA) is easy to adopt to cope with this kind of graphs, and from a cognitive science point of view, it seems similar with the behaviors of human snoopers. But the DFA seems not applicable in Web environment. Since usual Web pages are complicatedly inter-connected with other Web pages, it may bring about a long series of Web pages. The long series of Web pages may imply long time consumption to access a specific page.

On the other hand, there are several strengths in applying the breadth first algorithm (BFA) to Web site graphs. With the BFA an important Web page can easily be accessed. It is done by clicking relatively fewer steps from its homepage rather than by the DFA. It is easy to resolve a graph to a hierarchical tree and to minimize the depths to visit in a Web page. It can also be said that the access time can be minimized.

Wookey and Geller [14] suggested a weighted tree by a topological ordering algorithm that the tree is unbiased and minimize an average access from the root node. It also consider semantic relevance *(tf-idf)* between nodes in the same depth. But, when the Web page's weight is changed or the link structure is altered, total tree structure should be wholly reorganized.

3 Semantic Representation

3.1 Overview Visualization of a Web Sites

If an overview visualization of a Web site is additionally suggested to a user, it will be helpful to find a way where the user is. The tree structure applied by the breadth first algorithm is simple and easy to implement. But it is no use finding a significant page in a Web site or clustering pages with semantics. Thus we introduce an attribute called the weight to evaluate the significance of Web pages. Some experimental researches said that graphical representations support better navigation because this type of representation more precisely matches a user's mental model of the system [15, 16, 17]. Textual tools, however, suggest further advantages by allowing users to rapidly calibrate the extent of the site and to search visually in an efficient manner for particular information [18]. In this paper, we use the keyword-based measure as follows.

3.2 Evaluating the Weight Values

The weight for a Web page indicates how statistically important it is [19, 20, 21, 22]. One common way to compute a Web page (or document) W is the *tf-idf* that is first to obtain an unnormalized vector $W' = < w_1',...,w_m>^T$, where each w_i' (i = 1 to m, m is number of query terms) is the product of a word frequency (*tf*) factor and an inverse document frequency (*idf*) factor. The *tf* factor is equal (or proportional) to the frequency of the i^{th} word within the document. The *idf* factor corresponds to the content discriminating power of the i^{th} word: a 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 $log_2[N/df(q_i)]$, where N is the total number of documents, and $df(q_i)$ is number of documents containing the i^{th} word. If a word appears in every document, its discriminating power is 0. If a word appears in a single document, its discriminating power is as large as possible. Once W' is computed, the normalized vector W is typically obtained by dividing each w_i' term by $\sqrt{\sum_{i=1}^{m}(w_i')^2}$. The weight can be specified in this paper indicating the importance of the Web page.

3.3 Semantic Representation

If we can measure the weight of Web nodes corresponding to their significance, then the structure of the nodes can be manipulated by the weight. We introduce the *tf-idf* as the weight measure and it 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 previously described before, the *tf-idf* measure is applied as a weight of the Web node. The weight of Web node W_i corresponding to query vector Q composed of each query term q_i (i = 1 to m) is defined as a scalar derived as the inner product of the query vector Q with the Web page vector W_i:

$$\begin{aligned} weight(W_i) &= Q \bullet W' \\ &= [q_i] \bullet [tf_i \cdot \log_2[N/df(q_i)]]^T \\ &= \sum_i q_i \cdot (tf_i \cdot \log_2[N/df(q_i)]) \end{aligned} \quad \textbf{(2)}$$

The prototype system called AnchorWoman (ver. 1.4) has been tried to search for the structure of the test web site [23]. The link structure of the site refers Fig. 1.

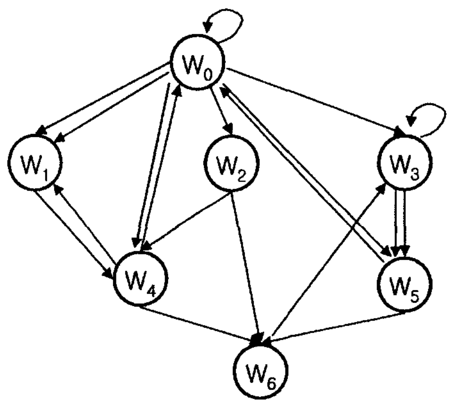
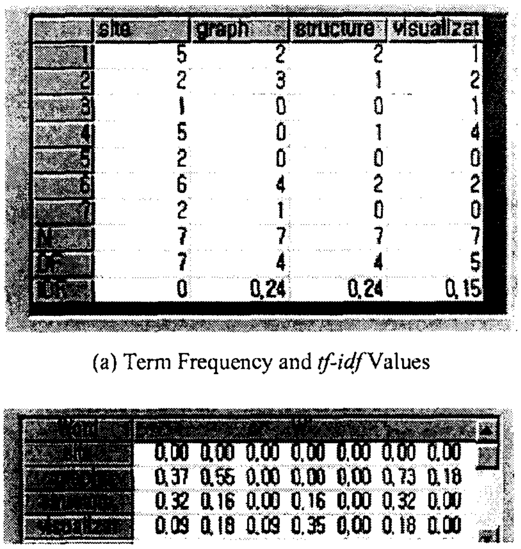


Fig. 1. Test Web site. A Circle represents a Web page, or a Web node and an arrow a hyperlink, or a web arc

The processes to extract the weight of keywords and the abstraction of Web sites are achieved by following (refer to the figures Fig. 2(a) and Fig. 2(b)). If we select four search terms 'site, graph,



	site	graph	structure	visualizat
1	5	2	2	1
2	2	3	1	2
3	1	0	0	1
4	5	0	1	4
5	2	0	0	0
6	6	4	2	2
7	2	1	0	0
N	7	7	7	7
DF	7	4	4	5
IDF	0	0,24	0,24	0,15

(a) Term Frequency and *tf-idf* Values

[illegible]	[illegible]						
[illegible]	0,00	0,00	0,00	0,00	0,00	0,00	0,00
[illegible]	0,37	0,55	0,00	0,00	0,00	0,73	0,18
[illegible]	0,32	0,16	0,00	0,16	0,00	0,32	0,00
[illegible]	0,09	0,18	0,09	0,35	0,00	0,18	0,00

(b) Normalized Weight Vector Generator

Fig. 2. Weight Production Module of AnchorWoman

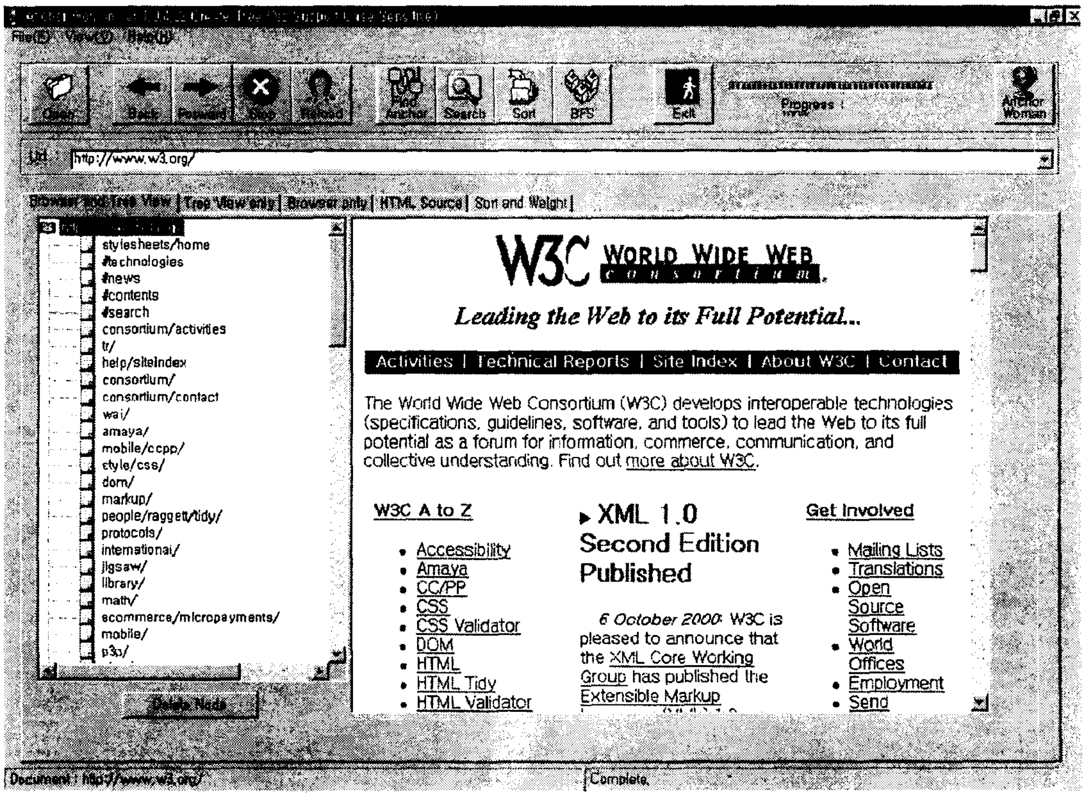
structure, visualization', then the weights are calculated as in Fig. 2(a). The word 'site', for example, appears 5 times in the homepage W_0 (the page number in the system appears as page 1), and in turn 2, 1, 5, 2, 6, and 2 times in W1.html to W7.html, respectively.

Finally, we get the normalized vectors as follows: W0 = <0, 0.37, 0.32, 0.09>, W1 = <0, 0.55, 0.16, 0.18>, ..., W7 = <0, 0.18, 0, 0> (Fig. 2(b)). As a similar process, we can obtain the weight vector W for the query word 'web' (of the Web test site). W = <9.49, 3.45, 6.21, 7.59, 9.66, 12.42>.

4 Description of System Requirement

4.1 Configurations

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



http://www.w3.org/

stylesheets/home
#technologies
#news
#contents
#search
consortium/activities
tr/
help/siteindex
consortium/
consortium/contact
wai/
amaya/
mobile/ccpp/
style/css/
dom/
markup/
people/raggett/tidy/
protocols/
international/
jigsaw/
library/
math/
ecommerce/micropayments/
mobile/
p3p/

W3C WORLD WIDE WEB consortium

Leading the Web to its Full Potential...

Activities | Technical Reports | Site Index | About W3C | Contact

The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential as a forum for information, commerce, communication, and collective understanding. Find out more about W3C.

W3C A to Z

- Accessibility
- Amaya
- CC/PP
- CSS
- CSS Validator
- DOM
- HTML
- HTML Tidy
- HTML Validator

▸ XML 1.0 Second Edition Published

6 October 2000: W3C is pleased to announce that the XML Core Working Group has published the Extensible Markup

Get Involved

- Mailing Lists
- Translations
- Open Source Software
- World Offices
- Employment
- Send

Fig. 3. An Example view of Web site (www.w3.org) with the AnchorWoman

browser, and a node weight tap. The system has been implemented with VB 6.0 as a client session and with MicroSoft ACCESS 2000 as a server database. The system begins with the homepage predetermined by the Web server and uses the interior anchors in the homepage. The anchors are classified with two categories such as interior anchors and exterior anchors. If the homepage begins with a frame without anchors, the system extracts anchors from which the HTML in the frame includes. The result anchors with page contents are stored in the database. The system, for example, finds that the Web test site's homepage includes three interior links (anchor).

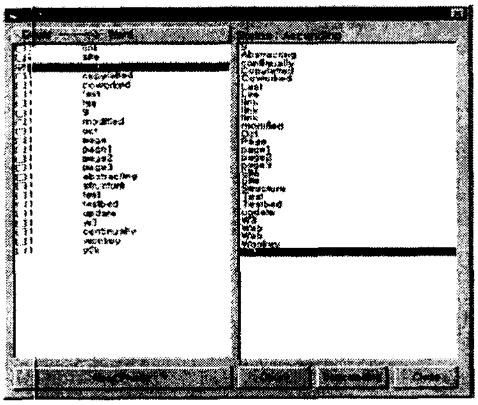
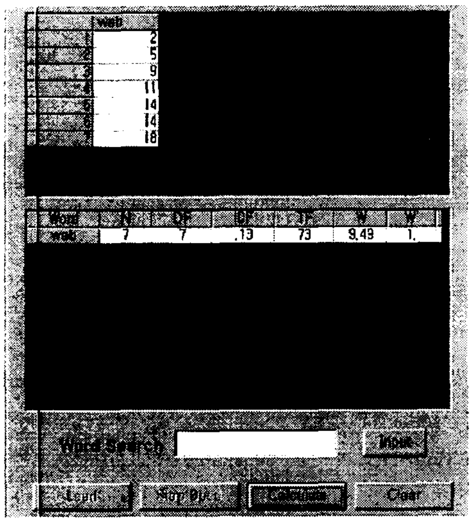


Fig. 4(a). Keyword selection module



	web
1	2
2	5
3	9
4	11
5	14
6	14
7	18

Word	N	DF	IDF	TF	W	W'
web	7	7	.13	73	9.49	1.

Fig. 4(b). Weight calculation in the system

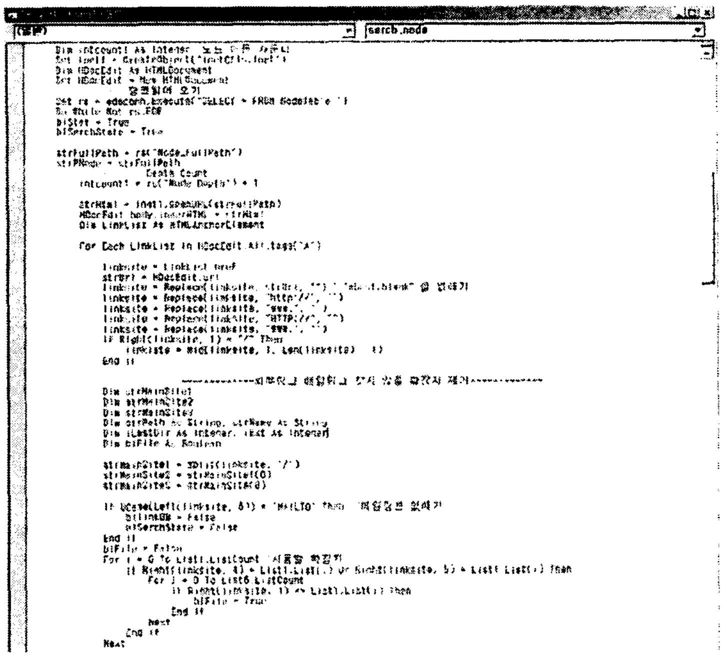
The anchors can be expanded in the site map of the system and the example is represented in the left-hand side of the Fig. 3. After getting the url (anchor and frame) extractor module, a web weight vector generator module in the AnchorWoman is represented in Fig. 4(a).

The system can generate the weights of Web pages relevant to the *tf-idf* measure. Users can also search specific keywords and generate the weights manually step by step. If we select a search term 'web', then the sum of the term is suggested in the upper side of the Fig. 4(b) and then the calculated weights in the below side of the Fig. 4(b). The word 'web', for example, is appeared 2 times in the homepage and 5, 9, 11, 14, 14, and 18 times through 1.html to 6.html respectively.

4.2 Preprocessing

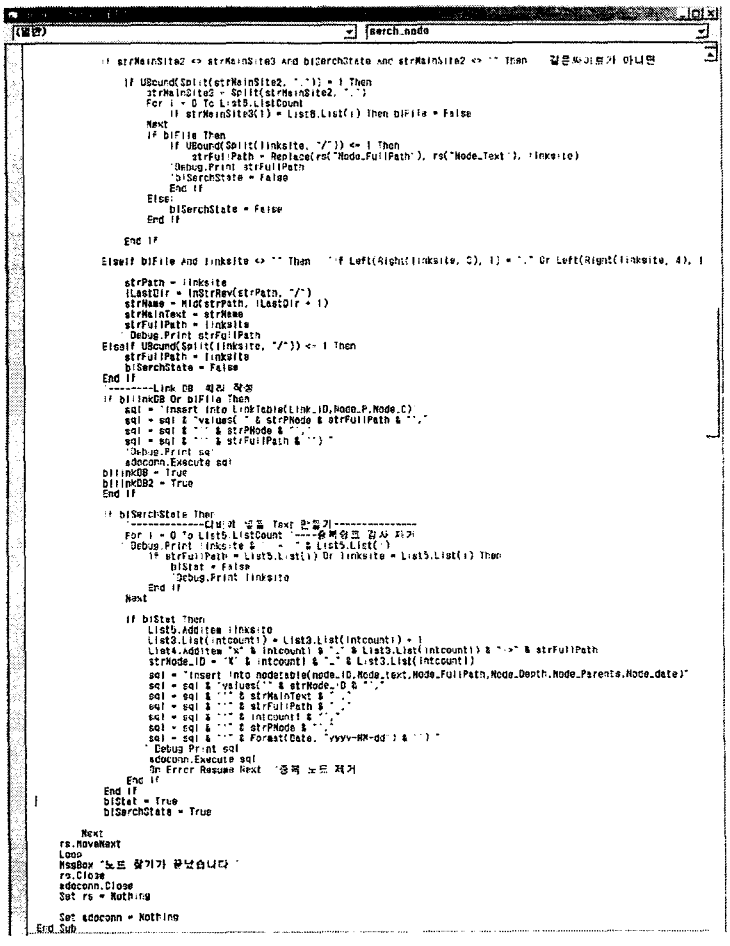
The AnchorWoman at first parses a given web site by the preprocessing procedure and extracts the corresponding Web information into a server database. Fig. 5(a) and Fig. 5(b) represent core modules for the AnchorWoman's preprocessing procedure to search the Web node with which the system produces the subordinate information in the database.

In the server database, the schemata for NodeTable and LinkTable 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 LinkTable are "Link_ID", "Parent Node," and "Child Node" respectively.



```
Dim intcount1 As Integer  '노드 깊이 카운터
Set Inet1 = CreateObject("InetCtls.Inet")
Dim HDocEdit As HTMLDocument
Set HDocEdit = New HTMLDocument
              '링크 읽어 오기
Set rs = adoconn.Execute("SELECT * FROM NodeTable ")
Do While Not rs.EOF
blStat = True
blSerchState = True

strFullPath = rs("Node_FullPath")
strPNode = strFullPath
              'Depth Count
    intcount1 = rs("Node_Depth") + 1

    strHtml = Inet1.OpenURL(strFullPath)
    HDocEdit.body.innerHTML = strHtml
    Dim LinkList As HTMLAnchorElement

    For Each LinkList In HDocEdit.All.tags("A")

        linksite = LinkList.href
        strUrl = HDocEdit.url
        linksite = Replace(linksite, strUrl, "") '"about:blank" 을 없애기
        linksite = Replace(linksite, "http://", "")
        linksite = Replace(linksite, "www.", "")
        linksite = Replace(linksite, "HTTP://", "")
        linksite = Replace(linksite, "WWW.", "")
        If Right(linksite, 1) = "/" Then
            linksite = Mid(linksite, 1, Len(linksite) - 1)
        End If

                   '-------------외부링크 해당하고 찾지 않을 확장자 제거-------------
        Dim strMainSite1
        Dim strMainSite2
        Dim strMainSite3
        Dim strPath As String, strName As String
        Dim lLastDir As Integer, lExt As Integer
        Dim blFile As Boolean

        strMainSite1 = Split(linksite, "/")
        strMainSite2 = strMainSite1(0)
        strMainSite3 = strMainSite1(0)

        If UCase(Left(linksite, 6)) = "MAILTO" Then  '메일링은 없애기
            blLinkDB = False
            blSerchState = False
        End If
        blFile = False
        For i = 0 To List1.ListCount  '서블릿 확장자
            If Right(linksite, 4) = List1.List(i) Or Right(linksite, 5) = List1.List(i) Then
                For j = 0 To List6.ListCount
                    If Right(linksite, 1) <> List1.List(j) Then
                        blFile = True
                    End If
                Next
            End If
        Next
```

Fig. 5(a). AnchorWoman's Preprocessing part



```
        If strMainSite2 <> strMainSite3 And blSerchState And strMainSite2 <> "" Then   '같은싸이트가 아니면

            If UBound(Split(strMainSite2, ".")) = 1 Then
                strMainSite3 = Split(strMainSite2, ".")
                For i = 0 To List8.ListCount
                    If strMainSite3(1) = List8.List(i) Then blFile = False
                Next
                If blFile Then
                    If UBound(Split(linksite, "/")) <= 1 Then
                        strFullPath = Replace(rs("Node_FullPath"), rs("Node_Text"), linksite)
                    'Debug.Print strFullPath
                    'blSerchState = False
                    End If
                Else:
                    blSerchState = False
                End If

            End If

        ElseIf blFile And linksite <> "" Then   'If Left(Right(linksite, 3), 1) = "." Or Left(Right(linksite, 4), 1

            strPath = linksite
            lLastDir = InStrRev(strPath, "/")
            strName = Mid(strPath, lLastDir + 1)
            strMainText = strName
            strFullPath = linksite
           ' Debug.Print strFullPath
        ElseIf UBound(Split(linksite, "/")) <= 1 Then
            strFullPath = linksite
            blSerchState = False
        End If
        '---------Link DB 처리 작성
        If blLinkDB Or blFile Then
            sql = "Insert into LinkTable(Link_ID,Node_P,Node_C)"
            sql = sql & "values( " & strPNode & strFullPath & "',"
            sql = sql & "'" & strPNode & "',"
            sql = sql & "'" & strFullPath & "') "
            'Debug.Print sql
            adoconn.Execute sql
        blLinkDB = True
        blLinkDB2 = True
        End If

        If blSerchState Then
            '-------------다음의 샘플 Text 만들기---------------
            For i = 0 To List5.ListCount '----중복링크 검사 하기
           ' Debug.Print linksite & "  -  " & List5.List(i)
                If strFullPath = List5.List(i) Or linksite = List5.List(i) Then
                    blStat = False
                    'Debug.Print linksite
                End If
            Next

            If blStat Then
                List5.AddItem linksite
                List3.List(intcount1) = List3.List(intcount1) + 1
                List4.AddItem "K" & intcount1 & "_" & List3.List(intcount1) & "->" & strFullPath
                strNode_ID = "K" & intcount1 & "_" & List3.List(intcount1)
                sql = "Insert into nodetable(node_ID,Node_text,Node_FullPath,Node_Depth,Node_Parents,Node_date)"
                sql = sql & "values('" & strNode_ID & "',"
                sql = sql & "'" & strMainText & "',"
                sql = sql & "'" & strFullPath & "',"
                sql = sql & "'" & intcount1 & "',"
                sql = sql & "'" & strPNode & "',"
                sql = sql & "'" & Format(Date, "yyyy-MM-dd") & "') "
               ' Debug Print sql
                adoconn.Execute sql
                On Error Resume Next  '중복 노드 제거
            End If
        End If
        blStat = True
        blSerchState = True

    Next
rs.MoveNext
Loop
MsgBox "노드 찾기가 끝났습니다 "
rs.Close
adoconn.Close
Set rs = Nothing

Set adoconn = Nothing
End Sub
```

Fig. 5(b). AnchorWoman's Preprocessing part

The source codes in Fig. 5(a) and 5(b) in detail represent that HTMLDocument Object and its methods, i.e., "Alltags("A")", "OpenURL('URL') find the anchor tags(<a>) from the web pages parsed by this procedure. Each anchor tag whose target link address is parsed to identify internal links [13, 14]. The target link addresses are incorporated into the NodeTable and the LinkTable 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 into the NodeTable and the LinkTable.

4.3 Functionalities for IP solver

We use LINDO 6.1(LINDO API 3.0) as a tool for solving IP model in the AnchorWoman system that the LINDO API functions get the data set from the VB application. 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 gets the data in the previous preprocessing procedure. The IP solver runs the core function of LINDO and generates the optimal solution upon the IP problem and to be used to IP reporter for sensitivity analyses interactively maintaining the client session.

5 An IP Approach

5.1 IP Modeling

It can be used in IP(Integer Programming) to convert a digraph to a tree. First, a tree possesses its own property to which an IP's constraints can be derived.

e objective function is to maximize the summation of the weight of nodes.

›de's indegree shoud be 1. (Except for the root node)

cycle should be removed. (Cycle detection algorithm will be explained later)

self-cycle within a node should be removed.

ıplicated paths between adjacent two nodes should be removed.

So, considering to above constraints, IP formulation is as followed.

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

$$s.t. \ \sum_{i,j \in N} x_{ij} \begin{cases} = 0 & j = 0 \\ \leq 1 & j \neq 0 \end{cases} \quad \forall i \qquad (3)\text{-}2$$

$$x_{ij_1} + \sum_{k=1}^{m-1} x_{j_k j_{k+1}} + x_{j_m i} \leq m \quad for\ m \geq 2 \qquad (3)\text{-}3$$

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

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

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

A directed Web graph G(N, A) consists of a Web node N and a Web link between Web nodes, where a Web node (called node) corresponds to a Web page and a Web link (called link) a hypertext link. A Web site is defined as a directed graph G **(W, E)** consisting of a finite Web node bag **W** and a finite Web arc bag **E** of ordered pairs of Web nodes. W and **E** are represented as a bag of Web node elements Wi and a bag of Web arc elements (Wi, Wj) respectively, where i, j = {0, 1, 2, 3, ... , n-1}, n represents cardinality of web pages =|**W**|, and a Web node set. Then the variable x_{ij} is 0 (when a Web link from node *i* to node *j* does not exist) or 1 (link from node *i* to node *j* exists). The parameter w_{ij} represents an average weight from node *i* to node *j*. There are several alternatives to derive the weight of a node and to generate a geometric distance to the link, including the number of inward or outward links [24]. Of course, it is not restricted to the method to generate a distance between the Web nodes. In this paper, we get a weight by *tf-idf* to each node, and generate a Euclidian distance w_{ij} from node *i* to node *j*.

The objective function 3-(1) means maximization of tree path's total sum of average mean weight. The constraint 3-(2) means each tree node's *indegree* should be 1 (except for the root node). The constraints 3-(3) and 3-(4) are to remove a cycle and a self-cycle respectively. The constraint 3-(6) represents the problem is a sort of IP(Integer Programming). That is to say, 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 survived in the result tree.

The cycle detection algorithm plays a role to detect cycle in digraph, and make topological order of node

```
Algorithm Cycle Detection
input
N,A  G(N,A) a directed graph
A(i) = adjacency list
procedure
{
    for (all iN) indegree(i)=0 do
    for (all (i,j)A) indegree(j)=indegree(j)+1 do
    LIST =  next = 0;
    for (all iN) 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 G contains a directed
    cycle
        else the graph G is acyclic and the result gives a topological order of its nodes;
}
```

in digraph. It is used in making *restriction 3-(3)*. Let a directed graph G=(N,A) has n node set N and m arc set A, and A(i) an adjacency list. The *indegree* is the number of Web arcs that ender into a Web page, the *outdegree* is the number of Web arcs that go from Web page. The LIST is a data structure that stores the current object that represents a Web page. The cycle detection algorithm is represented as above.

5.2 Sensitivity Analysis

When the query terms are altered, *tf-idf* measures of the Web Node are also altered. In this case, sensitivity analysis can be used in determining whether whole problem should be reformulated and recalculated or not. The standard IP problem form separated basic variable between nonbasic variable likes follows 4-(1), 4-(2), we can get equation $c_{\mathrm{BV}}x_{\mathrm{BV}} + c_{\mathrm{BV}}\mathrm{B}^{-1}\mathrm{N}x_{\mathrm{NBV}} = c_{\mathrm{BV}}\mathrm{B}^{-1}b$ by multiplying $c_{\mathrm{BV}}\mathrm{B}^{-1}$to constraint 4-(2).

$$
\begin{array}{llr}
z - c_{BV}x_{BV} - c_{NBV}x_{NBV} = 0 & & (4)\text{-}1 \\
st \quad Bx_{BV} + \quad Nx_{NBV} = b & & (4)\text{-}2 \\
\qquad\qquad x_{BV}, x_{NBV} \geq 0 & &
\end{array}
$$

Now, we can get the equation $c_{\mathrm{BV}}x_{\mathrm{BV}}= c_{\mathrm{BV}}\mathrm{B}^{-1}b - c_{\mathrm{BV}}\ \mathrm{B}^{-1}\mathrm{N}x_{\mathrm{NBV}}$. And substitute this equation for $c_{\mathrm{BV}}x_{\mathrm{BV}}$ term in 4-(1). then,

$$
\begin{array}{lr}
z - (c_{BV}B^{-1}b - c_{BV}B^{-1}Nx_{NBV}) - c_{NBV}x_{NBV} = 0 & (5)\text{-}1 \\
z + (c_{BV}B^{-1}N - c_{NBV})x_{NBV} = c_{BV}B^{-1}b & (5)\text{-}2
\end{array}
$$

The criteria of optimality in 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) is non negative. Also, the feasibility condition of the current basis solution is that equation (x_{BV}+$\mathbf{B}^{-1}\mathbf{N}x_{\mathrm{NBV}}$= $\mathbf{B}^{-1}\mathbf{b}$ which obtained by multiply equation 4-(2) by $\mathbf{B}^{-1}$)'s RHS is nonnegative. In other words, sufficient and necessary condition of current solution's optimality is as follows [25].[1)]

$\bar{c}_j = C_{BV}B^{-1}a_j - c_j \ \ 0$ (optimality condition, dual feasible condition)

$^{-1}$b (primal feasible condition)

When the weight of the Web node is changed, if these change does not influence the above two conditions (i.e., the optimality and feasibility condition) current basis is conserved. Details are explained in chapter 5.

6 Motivating Examples and Comparison

6.1 Motivating Example

Fig. 6. represents a digraph consist of 7 Web pages from Web page W_0 to Web page W_6. The weight of a arc is the mean average value (*tf-idf*) of two terminal nodes of the arc. By using the IP problem consists of equations from (3)-1 to (3)-6, we can get the solution in Fig. 8. (d).

[1)] Besides above two conditions, complementary slackness condition should be added. But, the Gaussian elimination in the Simplex Algorithm always keeps complementary slackness.

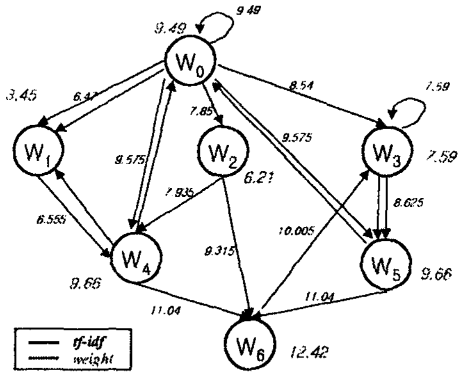


Fig. 6. An example Web site and its hierarchical abstraction using IP

After reorganize the IP problem to standard form like (4)-1 and (4)-2, each of BV(Basic variable), NBV(Nonbasic variable), objective *function* coefficient c_{BV}, c_{NBV}, basis matrix B, RHS(Right Hand Side) is described at Table 3 in appendix.

The alteration of IP problem induced by Web page's modification is represented as Table 1.

6.1.1: The alteration of the Web page's *tf-idf*

If Web page's *tf-idf* value is changed, corresponding objective function's coefficients of the variables are also changed. But, it don't break primal feasibility condition, i.e. $B^{-1}b$ So, if all nonbasic variable's $\tilde{c}_j = C_{BV} \cdot B^{-1} a_j - c_j$ is nonnegative then current basis does not change. Consider the case that W_4 value changed current 9.66 into 9.66+θ. 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

Table 1. The relationship between alteration of Homepage and IP standard form

	The modification in Web page	The change of IP standard form
Topology is conserved	The alteration of the Web page's *tf-idf*	Alteration of objective coefficient (c_{BV}, c_{NBV})
Topology is modified	The insertion of the new link (previous pages conserve)	Alteration of matrix **N** (Nonbasis coefficient aij for constraints ($x_{ij} = 0 \quad \forall i, j \notin N \quad$ ....3-(4)) that represent newly inserted link is changed from 1 to 0)
	The deletion of the link (previous pages conserve)	Insertion of the constraints (constraint $x_{ii} = 0 \quad \forall i, j \notin N \quad$ ....3-(4))
	The insertion of the 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

Then, the objective *function*'s coefficient that is changed now does not affect basis matrix Band 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 current basis does not change.

First, $C_{BV}B^{-1}$ is changed as follows.

$C_{BV}B^{-1}$= [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)[2]

= [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]

so,

$C_{BV}B^{-1}$= [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, 0.95, -0.085-Θ/2, -6.47, -7.85, -2.415, -9.575-Θ/2, -8.625, -11.04, -11.04, -3.105-Θ/2, -7.85, -10.005, -8.625, -11.04-Θ/2, -1.725-Θ/2, -6.47, -10.005, -8.625, -1.035-Θ/2, -5.47, -7.85, -9.575-Θ/2, -1.035-Θ/2, 0.085+Θ/2, 0.085, -10.005, -8.625, -6.47, -7.85, -1.38, -9.575-Θ/2, -9.575-Θ/2, -6.47, 1.465, 1.465-Θ/2, 2.415]

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 as follows.

$\bar{c}_{x03} = C_{BV}B^{-1}a_{x03} - c_{x03}$

= $C_{BV}B^{-1}$[000100000010000000000000000000 0000000000000000000000]^T-8.54

= 10.005-8.54=1.465≥0........................(1)

$\bar{c}_{x14} = C_{BV}B^{-1}a_{x14} - c_{x14}$

= $C_{BV}B^{-1}$[000010010000000000000000000000 0000000000000000000000]^T- 6.555

= 9.575+θ/2-6.555≥0............................(2)

$\bar{c}_{x24} = C_{BV}B^{-1}a_{x24} - c_{x24}$

= $C_{BV}B^{-1}$ [000010001000000000000000000000 0000000000000000000000]^T- 7.935

= 9.575+θ/2- 7.935≥0..........................(3)

$\bar{c}_{x26} = C_{BV}B^{-1}a_{x26} - c_{x26}$

= $C_{BV}B^{-1}$ [000000100000000000000000000000 0000000000000000000000]^T- 9.315

= 11.04+θ/2- 9.315≥0..........................(4)

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

= $C_{BV}B^{-1}$ [000100000000000000000000000000 0000000001000000000000]^T- 0

= 10.005-10.005-0≥0...........................(5)

$\bar{c}_{x50} = C_{BV}B^{-1}a_{x50} - c_{x50}$

= $C_{BV}B^{-1}$ [100000000010010000000000000000 0000000000000000000000]^T- 9.575

= 9.575+θ/2+0+0.95- 9.575≥0.............(6)

$\bar{c}_{x56} = C_{BV}B^{-1}a_{x56} - c_{x56}$

= $C_{BV}B^{-1}$ [000000100010000000000000000000 0000000000000000000000]^T- 11.04

= 11.04+θ/2-11.04≥0...........................(7)

$\bar{c}_{s1} = C_{BV}B^{-1}a_{s1} - c_{s1}$

= $C_{BV}B^{-1}$ [010000000000000000000000000000 0000000000000000000000]^T- 0

= 6.47- 0≥0...(8)

[2] The notation inv() means inverse matrix operator.

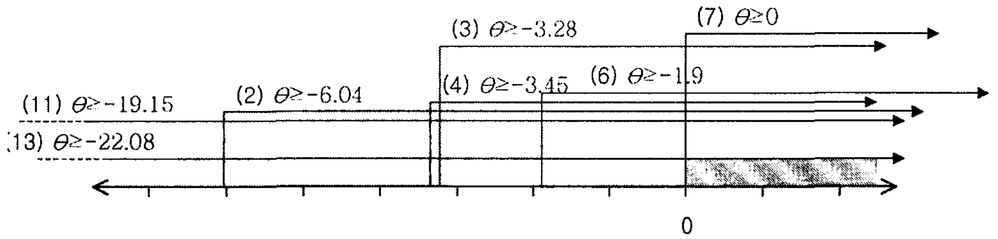


Fig. 7. Determination of range of Θ for which current basis remains optimal

$\bar{c_{s2}} = C_{BV}B^{-1}a_{s2} - c_{s2}$

$= C_{BV}B^{-1}$ [00100000000000000000000000
0000000000000000000]^T-0

$= 7.85- 0\geq0$..(9)

$\bar{c_{s3}} = C_{BV}B^{-1}a_{s3} - c_{s3}$

$= C_{BV}B^{-1}$ [00010000000000000000000000
0000000000000000000]^T-0

$= 10.005- 0\geq0$......................................(10)

$\bar{c_{s4}} = C_{BV}B^{-1}a_{s4} - c_{s4}$

$= C_{BV}B^{-1}$ [00001000000000000000000000
0000000000000000000]^T- 0

$= 9.575+\Theta/2- 0\geq0$...............................(11)

$\bar{c_{s5}} = C_{BV}B^{-1}a_{s5} - c_{s5}$

$= C_{BV}B^{-1}$ [00000100000000000000000000
0000000000000000000]^T- 0

$= 8.625- 0\geq0$..(12)

$\bar{c_{s6}} = C_{BV}B^{-1}a_{s6} - c_{s6}$

$= C_{BV}B^{-1}$ [00000010000000000000000000
0000000000000000000]^T- 0

$= 11.04+\Theta/2- 0\geq0$...............................(13)

$\bar{c_{s12}} = C_{BV}B^{-1}a_{s12} - c_{s12}$

$= C_{BV}B^{-1}$ [00000000000010000000000000
0000000000000000000]^T- 0

$= 0.95- 0\geq0$..(14)

Considering all range of (1)~(14), the result can be seen as Fig. 7. So, current basis is maintained as long as $\Theta\geq0$. In other words, although W$_4$'s *tf-idf* increases infinitely, current basis isn't affected by that change.

6.1.2: The insertion of the new link

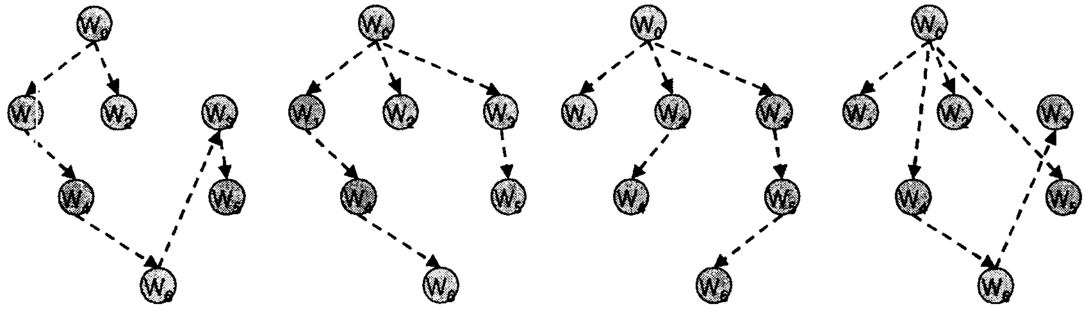
Consider the case of node W$_4$ and W$_3$ is connected (from W$_4$ to W$_3$). In matrix N, coefficient of variable x_{43}, i.e., a_{43}'s value is changed from 1 to 0. In this case, x_{43} is a nonbasic variable, so nonbasic matrix N's change does not hurt feasibility($\mathbf{B}^{-1}\mathbf{b}\geq0$) 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.

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

$= C_{BV}B^{-1}$ [00010000000000000000000000000
00000000000000000]^T- 0

$= 10.005-0\geq0$

In this case, current solution's feasibility condition and optimality condition is conserved. So, current basis does not changed.



Total Weight = 6.47+7.85+6.555+11.04+10.005+8.625 = 0.545

TW = 49.08

TW = 50.46

TW = 54.515

(a) DFS

(b) BFS

(c) WG

(d) IP

Fig. 8. The solution comparison to IP and others

6.1.3: The deletion of the link

In case of link deletion, the constraint that forms as $x_{ij} = 0 \quad \forall i, j \notin N$ ....3-(4) is inserted into IP formulation. Now, if current solution does satisfy the new constraint, current basis is maintained. In other case, dual simplex algorithm can be used.[3] Consider the case where two links between node W_1 and W_4 are disconnected. Then it should be included the two constraints(x_{14}=0 and x_{41}=0) into previous IP formulation. Because the current solution satisfies the inserted constraints, current basis is maintained.

6.1.4: The insertion of the new Web page

In this case, the objective *function* coefficient, the element of N matrix for new inserted variable, and all constraint type((3)-1~(3)-4, except nonnegative constraint (3)-5) can be included in IP formulation. So, this case's current basis does not conserved.

6.2 The Comparison to the other methodologies

Fig. 8 represents trees constructed by an IP and other methodologies. First, the tree that is constructed by DFS(Depth First Search) and BFS(Breath First Search) can easily be constructed by only traversing the digraph's link structure. But, DFS produces a left(or right) biased tree because it uses only the digraph's link structure and its depth first search strategy. In this case, it ignores Web page's semantic priority or a relationship between the nodes so that it may lead to user's wrong web traversing. In BFS, it makes a tree that minimizes an average access step from the root node to each node. But, it also ignores Web page's semantic priority, so the result tree can be semantically irrelevant.

The abstraction of a Web site can be said to lose the representative richness of hypertext links. But it may be said that it makes a direct access path for each Web pages. For once a (hierarchical) structure of a Web site is constructed, then any page can be

[3] In simplex algorithm, during keep feasibility (RHS≥0), optimality($\bar{c}_i = C_{BV} B^{-1} a_i - c_i \geq 0$) condition which currently does not satisfied is satisfying by pivot operation. But, In dual simplex algorithm, during keep optimality condition, feasibility condition which currently does not satisfied is satisfying by pivot operation.

accessed just with jumping (or clicking) to the specific node in the tree. Fundamentally there is, of course, a trade-off between the full generation method and an abstraction method. Web designer can limitedly support these kinds of needs [19]. We have rather a stress on the user side's viewpoint that a user wants his/her own needs on snooping a specific Web site.

The tree that is constructed by Wookey and Geller [14] is unbiased and minimize an average access from the root node to each nodes like BFS. It also considers semantic relevance(*tf-idf*) between nodes in the same depth. But, when the Web page's weight is changed or the link structure is altered, total tree structure should be wholly reorganized. Especially, because its own tie-breaking scheme, when Web page's *tf-idf* value is infinitesimally altered and previously, tie is occurred including that page, the tree structure should be changed though variation is infinitesimal.

Unlike other methods, because it considers semantic relevance among Web pages in the digraph, the tree constructed by IP can lead user's right web traversing. In Fig. 8, W_3's depth is 1 so that it can be accessed from the root node directly in the digraph. But, semantic relevance between W_3 and W_6 is higher rather than that of distance between W_1 and W_3, so W_3 is connected through W_6instead of the root node in the result tree. Also, it is self-evident that the tree is constructed by IP maximize total tree weight (In this example, total weight of IP tree is 54.515. See Table 2.). In respect of the tree depth or the number of span, the IP tree also represents superior result than the others. In DFS, due to the DFS strategy, maximum depth of the result tree mostly be larger than that of original web site. In case of BFS and WG[14], maximum depth of the result tree may be less than or equal to that of original web site structure. The method, however, can be inferior to that of the IP in the average depth measure because the result tree of IP is reconstructured based on *tf-idf* criteria and the number of span can be more reduced than these two methods. Finally, because IP methodology facilitates incremental tree construction from the current basis, it is useful to apply to Homepage that changes frequently.

Table 2. The result table of each methodology

	DFS	BFS	WG	IP
Total weight	50.545	49.08	50.46	54.515*
Depth				
Avg.	3	2	2	1.67*
Max.	5	3*	3*	3*
Min.	1*	1*	1*	1*
Span				
Max.	2	3	3	4

7 Conclusions

In this paper, we proposed the method that makes a virtual hierarchical tree structure of a given Web site in dynamic manner. By using further visual information like this, users can access web sites effectively and assess the whole web site range fast. Also, integer programming that we adopt can make website structure optimally. When Web site structure is changed frequently by applying sensitivity analysis, web site structure can keep stability.

References

1. Barabási 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
2. Conclin, J.: Hypertext, an introduction and survey.IEEE Computer 20(9) (1987) 17-41
3. Lau, T., Etzione, O., and Weld D. S.: Privacy interfaces for information management.Communications of the ACM 42(10) (1999) 89-94
4. Chi, E. H.: Improving Web Usability Through Visualization. IEEE Internet Computing 6(2) (2002) 64-71
5. Korfhage, R.: Information Storage and Retrieval. Wiley Computer Publishing (1997)
6. Thomas K. Landauer, Dennis E. Egan, Joel R. Remde, Michael Lesk, Carol C. Lochbaum, and Daniel Ketchum.: Enhancing the usability of text through computer delivery and formative evaluation: the SuperBook project. Hypertext: A Psychological Perspective. Ellis Horwood (1993) 71-136
7. Dennis E. Egan, Joel R. Remde, Louis M. Gomez, Thomas K. Landauer, Jennifer Eberhardt, and Carol C. Lochbaum.: Formative design evaluation of SuperBook. Transaction on Information Systems 7(1) (1989) 30-57
8. Dennis E. Egan, Joel R. Gemde, Thomas K. Landauer, Carol C. Lochbaum, and Louis M. Gomez.: Behavioral evaluation and analysis of a hypertext browser. In Proc of the ACM SIGCHI Conference on Human Factors in Computing Systems. May (1989) 205-210
9. Pandurangan, G., Raghavan, P. and Upfal, E.: Using Page Rank to Characterize Web Structure. Proc. Int'l Conference on COCOON (2002) 330-339
10. Pilgrim, C. J. and Leung, Y. K.: Designing WWW Site Map Systems. Proc. 10th Int'l Workshop on DEXA (1999) 253-258
11. Huang, M., Eades, P., Wang, J. and Doyle, B.: Dynamic Web Navigation with Information Filtering and Animated Visual Display. Proc. Int'l Conference on APWeb98 (1998) 63-71
12. Mendelzon, A. O. and Milo, T.: Formal model of web queries. Proc. ACM PODS (1997) 134-143
13. Ng, W. K., Lim, E. P., Huang, C. T., Bhowmick, S., and Qin, F. Q.: Web warehousing: An algebra for web information. Proc. of the IEEE ADL (1998) 228-237
14. L. Wookey, J. Geller: Semantic Hierarchical Abstraction of Web Site Structures for Web Searchers. Journal of Research and Practice in Information Technology, Vol. 36, No. 1, Feb. (2004) 71-82
15. Hasan, M. Z., Mendelzon, A. O. and Vista, D.: Applying database visualization to the world wide web. ACM SIGMOD RECORD 25(4) (1996) 45-49
16. Zwol, R. and Apers P.: The webspace method: On the integration of database technology with multimedia retrieval. Proc. CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, ACM Press (2000) 438-445
17. Risse, T., Leissler, M., Hemmje, M., Aberer, K. and Klement, T.: Supporting dynamic information visualization with VRML and databases. Proc. Workshop on New paradigms in information visualization and manipulation (1998) 69-72
18. Glover, E. J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D. M., and Flake G.: Using web structure for classifying and describing web pages. Proc. WWW (2002) 562-569
19. Chang, C. and Hsu, C.: Enabling Concept-Based Relevance Feedback for Information Retrieval on the WWW.IEEE TKDE, 11(4): July-Aug. (1999) 595-609
20. Chen, M., Hearst, M., Hong, J. and Lin, J.: Cha-Cha: A system for organizing intranet search results. Proc. USENIX Symposium on Internet Technologies and Systems, Boulder, CO, USA (1999) 11-14
21. Garvano, L., Garcia-Molina, H. and Tomasic, A.: GIOSS: Text-Source Discovery over the Internet. ACM TODS, 24(2): June (1999) 229-264
22. Garofalakis, J., Kappos, P. and Mourloukos, D.: Web Site Optimization Using Page Popularity, IEEE Internet Computing, 3(4): July-Aug (1999) 22-29
23. http://www.sungkyul.ac.kr/~wook/
24. Jingyu Hou and Yanchun Zhang.: Effective Finding Relevant Web Pages from Linkage Information.IEEE Trans. Knowledge and Data Engineering 15(4) (2003) 940-951
25. Katta G. Murty: Linear Programming. Wiley, New York. (1983)

Appendix:

Table 3. Standard from representation of example problem

	Basis	Nonbasis
Variable	$x_{BV}=[x_{40}, x_{01}, x_{04}, x_{63}, s_{8}, x_{35}, x_{46}, s_{7}, x_{02}, s_{9}, s_{10}, s_{11}, x_{05}, x_{00}, x_{11}, x_{22}, x_{33}, x_{44}, x_{55}, x_{66}, x_{06}, x_{10}, x_{12}, x_{13}, x_{15}, x_{16}, x_{20}, x_{21}, x_{23}, x_{25}, x_{30}, x_{31}, x_{32}, x_{34}, x_{36}, x_{41}, x_{42}, x_{43}, x_{45}, x_{51}, x_{52}, x_{53}, x_{54}, x_{60}, x_{61}, x_{62}, x_{64}, x_{65}]^{T}$	$x_{NBV}=[x_{03}, x_{14}, x_{24}, x_{26}, x_{43}, x_{56}, s_{1}, s_{2}, s_{3}, s_{4}, s_{5}, s_{6}, s_{12}]^{T}$
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 Matrix[4]	B=[100000000000010000000100001000100000000000010000 010000000000001000000000000100010001000100001000 000000001000000100000010000000001000100010000100 000100000000000010000001000010000000010001000000 001000000000000001000000000000000100000000100010 000001000000100000100000100001000000001000000001 000000100000000000011000010000000010000000000000 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 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000000000000 0000100000000 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]^{T}	

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