Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2008 Proceedings

International Conference on Information Systems (ICIS)

2008

Improving Web Site Structure to Facilitate Effective User Navigation

Min Chen
University of Texas at Dallas, mxc058000@utdallas.edu

Young Ryu
University of Texas at Dallas, ryoung@utdallas.edu

IMPROVING WEB SITE STRUCTURE TO FACILITATE EFFECTIVE USER NAVIGATION

Améliorer la structure du site Web pour faciliter la navigation efficace des utilisateurs

Min Chen
School of Management
University of Texas at Dallas
Richardson, Texas 75083
mxc058000@utdallas.edu

Young Ryu
School of Management
University of Texas at Dallas
Richardson, Texas 75083
ryoung@utdallas.edu

Abstract

Web sites are most effective when they meet both the contents and usability needs of their users. It is revealed, however, that designing usable Web sites is not a trivial task. A primary reason is that Web developers' perceptions and knowledge can be very different from those of the target users. Such differences result in cases in which users cannot easily locate the relevant information in a Web site. In this paper, we propose a math programming model to improve the navigation effectiveness of a Web site while preserving its original structure whenever possible. Our approach minimizes unnecessary changes to the present structure of a Web site and hence can be applied for Web site maintenance on a regular basis. Our test on a real Web site shows that the approach can provide significant improvements over the Web site structure by introducing only a small number of new links.

Keywords: Web site usability, navigation effectiveness, Web usage mining, integer programming

Résumé

Nous proposons un modèle de programmation mathématique afin d'améliorer l'efficacité de navigation d'un site Web tout en préservant, autant que possible, sa structure originale. Notre approche réduit au minimum les changements inutiles de la structure actuelle et par conséquent peut être appliquée pour la maintenance régulière du site Web. Notre test sur un vrai site Web montre que l'approche peut apporter des améliorations significatives de la structure de site Web en introduisant seulement un nombre restreint de nouveaux liens.

Introduction

Web sites are a very important intermediary between firms and customers. In order to cater for the increasing demands from users, firms are spending large sums of money developing and maintaining their Web sites. A JupiterResearch survey points out that nearly half of the surveyed corporations have two to four major Web site implementations planned for 2004, and almost one-quarter are prepared to spend at least $1 million (Greenspan 2004). Another survey conducted in 2006 indicates that respondents spend a mean 27% more on maintaining their Web sites than that in the previous year (Chiger 2006). With all these expenditures on Web site development and maintenance, it is still revealed, however, that finding desired information in a Web site is not simple (Dhyani et al. 2002; Otter and Johnson 2000) and designing usable Web sites is not a trivial task (Lazar 2006).

A primary cause of poor Web sites design is that Web developers' perceptions and knowledge can be different from those of the target users (Lazar 2006; Nakayama et al. 2000; Perkowitz and Etzioni 2000). Such differences result in cases in which users cannot easily locate the relevant information in a Web site. This is particularly true given the fact that many Web sites present navigation by using an organizational structure, but their users are not familiar with the structure. This kind of problem is difficult to avoid because when a Web designer creates a Web site, she may not have a clear understanding of the users' preferences and can only organize the pages based on her own judgment. However, the measure of effectiveness of the site should be the satisfaction of the users' expectations rather than that of researchers and designers (Marsico and Levialdi 2004). These scenarios point to a need to improve a Web site by mitigating such a perception and knowledge discrepancy between Web developers and users.

Our work is closely related to the literature of optimizing Web site structure through the use of user navigation data. There are two general ways to improve the browsing efficiency of a Web site through manipulating its structure (Perkowitz and Etzioni 2000): to facilitate a particular user by dynamically reconstituting pages based on the user profile and the user's traversal paths, often referred as *personalization*; and to modify the Web site structure to ease the navigation for all users, often referred as *transformation*. While both approaches rely on mining users access patterns from Web log files, the personalization approaches are computationally intensive because they require tracking past usage for each user and generating Web pages accordingly. Moreover, user delay due to the dynamic link reorganization may be excessive and intolerable in personalization approaches. The transformation approaches, on the other hand, make use of aggregate usage data mined from Web log files and optimize Web structure for all users. These approaches do not allow user-specific customization of links, and hence there is no need to collect information of users' past usage. Consequently, the computation resource required for transformation approaches is considerably less than that for personalization approaches. In this paper, we are primarily concerned with the transformation approaches, so our approach is mostly applicable to Web sites whose contents on a page do not change in response to different contexts or conditions.

The literature considering Web site transformation primarily focuses on optimizing the structure of a Web site by substantially reorganizing the links among Web pages. Fu et al. (2002) seek to build adaptive Web sites by evolving site structure to facilitate user access. Specifically, they describe an approach to reorganize Web sites that provide users with their desired information with fewer clicks. Lin (2006) develops integer programming models for reorganizing Web sites based on the cohesion between Web pages, which is obtained from Web usage mining as hit rates. The models reduce the information overload and search depth for users surfing in the Web site. In addition, a two-stage heuristic is proposed to reduce the computation time. Gupta et al. (2007) propose a heuristic scheme based on simulated annealing that makes use of the aggregate user preference data to re-link the pages to improve navigability. Yen (2007) recently conducts a study in which several accessibility models are proposed to measure the degree of easiness for users to retrieve the Web pages in navigation. A case study is performed on a Web site of an Electronic Commerce course to illustrate the application of the proposed accessibility models.

In this paper, we seek to improve a Web site structure from the perspective of Web site maintenance rather than to reorganize it substantially. Specifically, we propose an integer programming (IP) model to minimize unnecessary changes to a Web site while improving the navigation effectiveness of the site. We explain in detail the metric we use to evaluate the discrepancy in the next section. The studies on Web site structure optimization (Fu et al. 2002; Gupta et al. 2007) model the number of outward links allowable per page as a tight constraint, i.e., a Web page may not contain more than, for example, 20 outward links. We do not impose the number of outward links allowable per page as a constraint in our formulation. Instead, we choose to incorporate a term in our objective function to penalize the Web pages whose outward links exceed a given threshold value.

We make the following assumptions: (i) users have some information goal, i.e., some specific information they are seeking, when they visit a Web site (Gupta et el. 2007; Marsico and Levialdi 2004); (ii) the Web site designer wants to bridge the discrepancy between the Web site structure and users' expectation with minimal changes to the structure; (iii) a user will follow the path that appears most likely to lead her to target(s) (Srikant and Yang 2001).

There are several novelties in our work. We minimize the alteration to the present structure of a Web site such that the resulting site preserves business or organizational logics. We model the number of outward links allowed on each page as a penalty term in our objective function instead of as a constraint. This renders more flexibility to the structure of the resulting Web site. We use aggregate usage data and our optimization objective is for all users, so there is no need to track the past usage of each new user. The benefit of this is that we do not have to collect such information and we could avoid a potentially high cost needed to provide dynamic link reorganization within a tolerable user delay. Finally, our approach focuses on the change of the link structure, so it will not introduce new pages.

Metric for Improving Web Site Structure

This paper addresses the question of how to improve a Web site structure from the perspective of Web site maintenance. So the first question is, given a Web site, how to evaluate its effectiveness. ISO 9241-11 (1998) formally defines effectiveness as the accuracy and completeness with which users achieve specified goals. In our context, we use "navigation effectiveness" to refer to the easiness and correctness of users to locate their informational goal in a Web site. Palmer (2002) points out that a Web site is said to be easy-navigated when users can access desired data without getting lost or having to backtrack. Marsico and Levialdi (2004) also indicate that information becomes useful only if it is presented in a way consistent with the target users' expectation. We follow this idea and evaluate the navigation effectiveness of a Web site based on how consistent its information is organized to the expectation of the site visitors. Thus, a well-structured Web site should be organized in such a way that the discrepancy between how the Web site is actually structured and users' expectation of the structure is minimized. We measure this discrepancy with the metric based on the number of times a user attempted before successfully locating the target. The more a user attempts to find her target, the more efforts she exerts and the more discrepant the Web site structure is from the user's expected structure. Therefore, a well-designed Web site should allow users to locate targets in the least number of attempts, which also means a high accuracy according to the definition of effectiveness in the ISO 9241-11.

Users are assumed to have some specific targets when they visit a Web site. They will follow the path that is most likely to lead them to the target. A user faces two choices when continuing on the current path is unlikely to lead her to the target page: she either backtracks to an already visited page to attempt a new path, or she gives up at the current page. A user may choose either of these two actions with some probabilities. This is analogous to the notion of *information scent* developed in the context of *information foraging theory* (Pirolli and Card 1999). Information scent refers to link cues that are associated with snippets of text and graphics of links. Information foragers use the information scent to predict what they will find if they pursue a certain path through a Web site and make navigation decisions accordingly. This suggests that a user may attempt a new path if she could not find the target in the current one (or the information scent is weak). Each time a user fails to locate the target in the current path and starts a new path, it signifies that the user attempts one time to seek for the target. Therefore, we use the number of paths a user traversed to find the target as a proximity to the number of times that user attempted to locate the target.

We use backtracks to identify when a new path starts, where a backtrack is defined as a user's click on the "back" button in the browser or a revisit to an already visited page. The insight is that visitors will backtrack if they do not find the page where they expect it (Srikant and Yang 2001). In this sense, a *path* is defined here as a sequence of Web pages accessed by a user in which she does not revisit an already visited page. Chen et al. (1998) use *maximal forward reference* to denote such a path. Essentially, a maximal forward reference or a path is a sequence of pages a user traversed without backtracking. Thus, every backtracking point represents the end of one maximal forward reference. In this paper, we use the term path instead of maximal forward reference for simplicity. Note that the number of paths is always one greater than the number of backtracks. For example, if a user backtracks three times, then she must traverse four paths. Obviously, the more paths a user has attempted to reach her target, the more discrepant a Web site structure is from the user's expectation and the more likely that users will give up searching for targets.

We use an example to illustrate the above-mentioned concepts and how we can obtain paths from Web log files. Web log files store the information about page requests from users, including client IP address, request date/time, page requested, etc. To analyze the interaction between users and the Web site, the log files must be broken up into user *sessions*. A session is defined as a group of activities performed by a user during her visit to a site (Cooley et al. 1999). According to this definition, a session can include more than one target page as a user may access several targets during a single visit. Because the metric used in our analysis is the number of paths traversed by a user to locate one target, we use a different term *mini session* to refer to a group of pages visited by a user during her visit for *one* target. So, a session may comprise one or more mini sessions, each of which consists of a set of paths (maximal forward references). Many preprocessing algorithms and heuristics have been proposed to demarcate sessions from raw Web log files (Cooley et al. 1999; Spiliopoulou et al. 2003). In this paper, we identify if a page is a target by evaluating whether the time spent on that page is greater than a timeout threshold. The intuition is that a user generally spends more time on a page of her interests as compared to a page that is not her target. We note that this page-stay timeout heuristic, like other heuristics used in the literature, is nevertheless not perfect. Gupta et al. (2007) point out that it is not possible to identify user sessions unerringly from the user access log files.

Figure 1 shows a hypothetical Web site with ten pages; solid arrows represent existing links between pages. Note that the structure of the Web site is random and does not exhibit any special characteristics. Figure 2 illustrates a possible traversal path where a user starts from *A*, goes to *D* and *H*, then backtracks to *D*, from where she visits *C*, *B*, *E*, and *J*, and then backtracks to *B*. After that, this visitor goes from *B* to *F* and finally reaches *K*, the target.

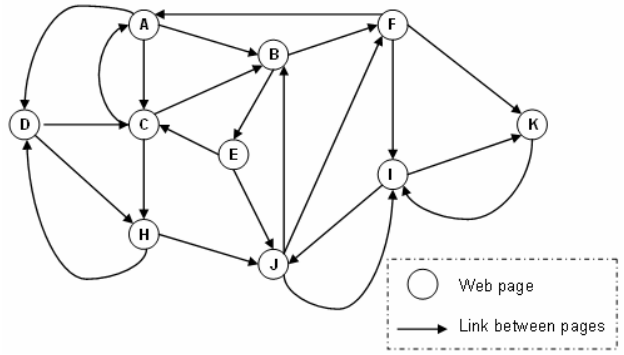

Figure 1. A Web Site With Ten Pages

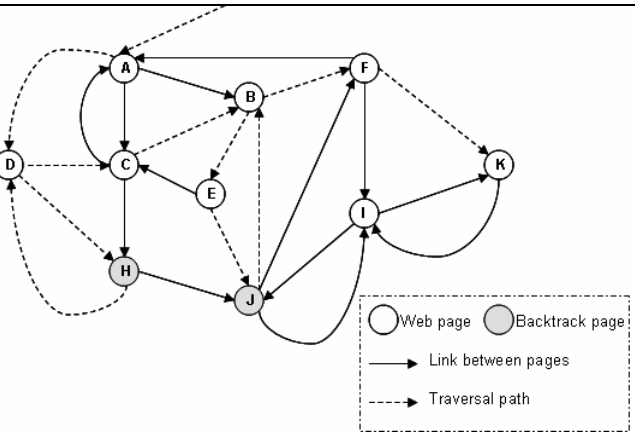

Figure 2. One Possible Traversal Path

If *K* is the only target in this traversal path, then this user backtracks at *H* and *J* before reaching *K*. Subsequently, all Web pages this user visited in the traversal path belong to one mini session and are ordered in the sequence of when

they are accessed. For example, the mini session in figure 2 can be denoted as $S = \{\{A, D, H\}, \{C, B, E, J\}, \{F, K\}\}$. Note that D and B only appear once in mini session S because of caching. This mini session has three elements and each element represents a path, which is set of pages a user visited without backtracking. For example, $\{C, B, E, J\}$, the second element of S, denotes the set of pages accessed in the second path in the mini session. Page K, the last access page in the last path, is the user's target page in this mini session.

This visitor has navigated three paths to search the target. In the first path, she visited A, D and H in sequence. In her second path, she traversed C, B, E, and J after she backtracks from H to D. In the last path, she backtracks to B and then reached the target K via F. A simple solution to help this visitor reach the target faster is to introduce extra links (Lin et al. 2002; Nakayama et al. 2000; Perkowitz and Etzioni 2000). There are many ways of adding extra links in our case. If a link is added from D to K, then this user could directly reach K via D and therefore reach the target in the first path, as compared to the original structure, in which she needs to traverse three paths based on her expectation of the site structure. Hence, we help her "save" two paths by linking D to K. Similarly, if we choose to link B to K, then this visitor will only backtrack once at H and then she could follow the link from B to K in the second path. In this case, she is said to save one path in the improved structure as compared to the original structure. If we choose to link E to K, the effect of saving the visitor's navigation path is considered the same as linking B to K, because both B and E are pages in the second path attempted by the user, linking them to K only allows the visitor to save one path. Still, another possibility is to link C to F, where neither of the two nodes is the target. In such a case, whether a visitor will follow the old link or the new link, which does not link directly to the target, is not intuitive. Since a visitor is assumed to follow the path that appears most likely to lead her to the target, she is assumed to follow the old path if a new link does not directly connect a page to the target.

Problem Formulation

This section introduces the formulation of the integer programming model for our Web site maintenance problem and discusses other relevant aspects. The Web site maintenance problem can be viewed as a generalization of the hitting set problem, which is known to be a difficult problem in general.

The Integer Programming Model

The Web Site maintenance problem can be regarded as a special graph optimization problem. We model a Web site as a directed graph, in which Web pages are represented as nodes and links among pages are denoted as arcs. Let N be the set of all pages. For every page $i, j \in N$ ($i \neq j$), expression $\lambda_{ij} = 1$ indicates there exists a link from page i to page j; expression $\lambda_{ij} = 0$ indicates there does not exist a link from page i to page j. Further, $\#_i$ denotes the total number of links originating from page i, $i \in N$. From the access log, we obtain the set T of all mini sessions. Define $E = \{(i, j)$: $i, j \in N$ ($i \neq j$), and $\exists S \in T$ such that i is a page of S and j is the target page of $S\}$. Our problem is to determine whether to establish a link from i to j for $(i, j) \in E$. Let $x_{ij} \in \{0, 1\}$ denote the decision variable such that $x_{ij} = 1$ indicates establishing the link.

For each mini session S, $S \in T$, $L(S)$ denotes the number of paths in S and $L(b,S)$, $b \leq L(S)$ denotes the number of Web pages visited in the bth path in mini session S, that is, the length of that path. In the example illustrated in figure 2, $L(S)=3$ because mini session S consists of three paths. In the first path, the visitor traversed page A, D and H, so $L(1,S)=3$. Similarly, $L(2,S)=4$, $L(3,S)=2$, indicating that 4 and 2 pages were visited in the second and third paths. Let $docno(m, p, S)$ be the mth Web page visited in the pth path in mini session S, $1 \leq p \leq L(S)$, $1 \leq m \leq L(p, S)$. Other notations used in this paper are listed as follows:

1. $$a_{ijkm}^{S} = \begin{cases} 1 \text{ if } docno(m,k,S) = i \text{ and } docno(L(L(S),S),L(S),S) = j \\ 0 \text{ Otherwise} \end{cases}$$

Therefore, $a_{ijkm}^{S} = 1$ if and only if Web page i is the mth page visited in the kth path in mini session S and page j is the target page in this session. The values of a_{ijkm}^{S} can be easily obtained from log file data in preprocessing steps.

2. b_j: the *path tolerance* value, defined as the maximum number of paths allowed in mini sessions containing j as the target. Since every backtrack can be viewed as the end of one path attempted by a user to find the target, a firm may want to ensure user satisfaction by limiting the number of paths in a mini session to a tolerable bound,

denoted as b_j. We will explain how to choose proper values for b_j later in this section.

3. Define $c_{km}^S = \sum_{(i,j)\in E} a_{ijkm}^S x_{ij}, m = 1...L(k,S), k = 1...L(S), \forall S \in T$

$$c_{km}^S = \begin{cases} 1 \text{ if the } m\text{th page in the } k\text{th path is selected to link the target in mini session } S \\ 0 \text{ Otherwise} \end{cases}$$

So $c_{km}^S = 1$ if and only if $a_{ijkm}^S = 1$ and $x_{ij} = 1$, for some $(i, j) \in E$. a_{ijkm}^S is used to establish correspondence between the global indices *i, j* and local indices *S, k* and *m*, as our objective is to minimize the number of new links which is denoted using global indices, but the constraint that ensures minimum user satisfaction is defined using local indices.

4. C_i: the out-degree threshold value, which is the maximum number of outward links deemed reasonable for page *i*.

The Web site maintenance problem (WEB-MAI) is formulated as follows:

$$\text{minimize} \quad \sum_{(i,j)\in E} x_{ij}(1-\lambda_{ij}) + m\sum_{i\in N} p_i$$

s.t.

$$\sum_{k=1}^{b_j} \sum_{m=1}^{L(k,S)} c_{km}^S \geq 1, \forall S \in T \tag{1}$$

$$\sum_{j:(i,j)\in E} x_{ij}(1-\lambda_{ij}) + \#_i - p_i \leq C_i, \forall i \in N_i, N_i = \{i : (i,j) \in E\} \tag{2}$$

$$\begin{aligned} &x_{ij} \in \{0,1\}, \forall (i,j) \in E, p_i \in \{0\} \bigcup Z^+ \\ &c_{km}^S \in \{0,1\}, m = 1...L(k,S), k = 1,2...L(S), S \in T \end{aligned} \tag{3}$$

The objective function minimizes the number of new links to be established and the penalties imposed to the nodes with excessive number of outgoing links. There are cases where the solution suggests linking two nodes where a link already exists, i.e., x_{ij}=λ_{ij}=1. This implies that visitors could have traversed from page *i* to target page *j*, but did not choose to do so in practice. This phenomenon suggests that improvements be made over the existing links instead of introducing new links. For example, the description of the current link may be misleading and confusing such that visitors may not realize that they can reach their targets via this link. It may also be the case that the current link is poorly designed or placed in an inconspicuous location so that many visitors who do not scan all available links may fail to find it. These phenomena are related to the issue of Web page content design. When they occur, it is unnecessary to have a duplicated new link introduced. Instead the Web designer is informed of these links and she should focus on improving the existing link to make it more effective. Therefore, we introduce $(1-\lambda_{ij})$ in the objective function to let the model pick the existing links whenever possible.

Constraint (1) states that for all mini sessions with target *j* and a path tolerance value b_j, at least one link to the target page must be established or improved from a Web page visited before the start of the (b_j+1)th path. In this sense, the path tolerance value in fact represents the minimum user satisfaction level imposed by the Webmaster. Constraint (2) captures, for page *i*, the number of links exceeding the outward link threshold C_i. p_i denotes the number of outward links in page *i* that exceeds the threshold C_i. Constraint (3) imposes that decision variables are binary and p_i is a non-negative integer. *m* is the multiplier for the penalty term in the objective function. The value of *m* depends on the relative harm to a Web site of having a new link in a page with less than C_i outward links as compared to adding a link in a page whose number of outward links exceeds the defined threshold.

The formulation states that visitors are able to reach their targets in no more than b_j attempts. Note that b_j could take different values for different targets. Take an E-commerce Web site for example. Some products may associate with high profits, so the firm may want them to be discovered in as few attempts as possible and Web pages of these products should have a lower b_j. In contrast, items with relatively low profits may tolerate a greater b_j. Note that

setting b_j=1 for all targets js allows visitors to locate their targets in only one path without any backtracking. It is worthwhile noting that the hitting set problem[1] can be viewed as a special case of (WEB-MAI) when m=0.

An Example

The entire Web site maintenance problem is illustrated here with an example. Table 1 shows 6 mini sessions, each of which consists of at least 2 paths, indicating that these visitors backtrack at least once. A path is represented as an element in the mini session and is a set itself, which comprises the set of pages that were accessed in that path. For example, S_1={{2, 1}, {4}, {3, 6}} denotes a mini session in which a visitor looks for page 6. She starts from page 2, clicks on page 1, then backtracks to page 2 and clicks on page 4. She could not find the target under page 4, so she again backtracks to page 2. In the last attempt, she traverses to page 3 and reaches page 6 via page 3. Among the 6 mini sessions, 3 sessions comprise exactly 3 paths, 2 sessions have only 2 paths and 1 session contains 4 paths. Page 6 is the target of 3 sessions (S_1, S_2, and S_5) and page 4 is the target of the remaining 3 sessions.

Table 1. Example Mini Sessions

ID	Mini Sessions
S_1	{{2, 1}, {4}, {3, 6}}
S_2	{{4, 3},{5, 2}, {1, 6}}
S_3	{{1, 5, 3}, {2, 4}}
S_4	{{6, 3}, {2, 5}, {1, 4}}
S_5	{{4, 1}, {5}, {2}, {3, 6}}
S_6	{{5, 3, 1}, {2, 4}}

Suppose that the following is the connectivity matrix of the six pages in our example:

$$\begin{pmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \\ p_1 & 0 & 0 & 0 & 1 & 1 & 1 \\ p_2 & 1 & 0 & 1 & 1 & 1 & 0 \\ p_3 & 1 & 1 & 0 & 0 & 0 & 1 \\ p_4 & 1 & 0 & 1 & 0 & 1 & 0 \\ p_5 & 1 & 1 & 1 & 0 & 0 & 1 \\ p_6 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

The information of existing links can be intuitively obtained from this matrix. For example, the connectivity matrix shows that page 1 has a link to page 5, i.e., λ_{15} =1, but does not contain a link to page 3, i.e., λ_{13} =0. For the purpose of illustration, we do not consider the penalty term in the objective function. As an example, suppose that the Webmaster wants to improve the site structure so that visitors are able to reach their targets in only 1 path. That is, for each mini session, we need to establish or improve at least one link from the set of pages accessed in the first path to the target page. Table 2 lists the potential links that could be introduced to accomplish this task.

[1] The hitting set problem is stated as follows: Given a ground set T and a collection of subset C, the goal is to find the smallest subset $H \subseteq T$ of elements that "hits" every set of C, i.e., $H \cap S \neq \emptyset$ for every $S \in C$. It is the dual to the set-covering problem and is known to be NP-complete (Garey and Johnson 1979)

Table 2. Potential Links When Path Tolerance *b*=1	
ID	Potential Links
S_1	{(2, 6), (1, 6)}
S_2	{(4, 6), (3, 6)}
S_3	{(1, 4), (5, 4), (3, 4)}
S_4	{(6, 4), (3, 4)}
S_5	{(4, 6), (1, 6)}
S_6	{(5, 4), (3, 4), (1, 4)}

The Web site maintenance formulation is

$$\text{minimize} \quad \sum_{i=1}^{6} \sum_{j=1, j\neq i}^{6} x_{ij}(1-\lambda_{ij})$$

s.t.

$$c_{11}^{S_1} + c_{12}^{S_1} \geq 1, \qquad (S_1)$$
$$c_{11}^{S_2} + c_{12}^{S_2} \geq 1, \qquad (S_2)$$
$$c_{11}^{S_3} + c_{12}^{S_3} + c_{13}^{S_3} \geq 1, \qquad (S_3)$$
$$c_{11}^{S_4} + c_{12}^{S_4} \geq 1, \qquad (S_4)$$
$$c_{11}^{S_5} + c_{12}^{S_5} \geq 1, \qquad (S_5)$$
$$c_{11}^{S_6} + c_{12}^{S_6} + c_{13}^{S_6} \geq 1, \qquad (S_6)$$
$$x_{ij} \in \{0,1\},\ i, j = \{1,2,3,4,5,6\}$$

where $c_{km}^{S} = \sum_{(i,j)\in E} a_{ijkm}^{S} x_{ij}, m = 1...L(k,S), k = 1...L(S), \forall S \in T$. This term establishes connections between c_{km}^{S} and x_{ij}, so that when x_{ij}=1 and $a_{ijkm}^{S} = 1$, we have its corresponding $c_{km}^{S} = 1$. For example, if x_{16}=1 is in the optimal solution and we obtain $a_{1612}^{S_5} = a_{1612}^{S_1} = 1$ from the data, then correspondingly we have $c_{12}^{S_5} = c_{12}^{S_1} = 1$ in the constraint, indicating that having a link from page 1 to page 6 is equivalent to having a link from the second page in the first path to the target page in mini sessions S_1 and S_5. The value of a_{ijkm}^{S} can be easily obtained from the data. For example, we have $a_{2611}^{S_1} = 1$ in the above example to denote that the first element of the first path in mini session S_1 is page 2 and the target is page 6. The other values can be obtained similarly.

Solving this integer program results in the optimal solution x_{36}=x_{34}=x_{16}=x_{14}=1, with the other variables being 0. Because we have λ_{16}=λ_{36}=λ_{14}=1 from the connectivity matrix, only the link (3, 4) needs to be established to improve the 6 mini sessions in our example. The other links, i.e., (1, 6), (1, 4) and (3, 6), will be reported to Web designers for improvements. For example, the visitor in S_1 could have traversed to page 6 while she was in page 1, but she chose to backtrack and clicked on other pages instead.

Computational Experiments and Results

Description of the Data Set

We used log data from the Music Machines Web site (http://machines.hyperreal.org). We chose this data set because it is publicly available and it has been widely used in the literature, e.g., (Perkowitz and Etzioni 2000, Gupta et al. 2007). Table 3 shows the number of Web pages that had out-degrees within a specified range. Out of 916 pages, 716 have an out-degree of no more than 20 and the majority of the remaining pages have fewer than 40 outward links.

Table 3. Out-degree Statistics

Out-degree	Number of Nodes
>100	1
81–100	2
61–80	10
41–60	21
21–40	166
11–20	538
5–10	44
1–4	47
0	87

The log file data used in our experiment contains 4,136,104 requests that were recorded from January 1999 to April 1999. Web logs usually contain some irrelevant information and cannot be directly used before being processed. We followed the standard log preprocessing steps described in (Fu et al. 2002) to filter the irrelevant information. We then used a Java application called the Web Utilization Miner (WUM) to analyze the log file. We utilized the page-stay time to distinguish target pages from other pages and to demarcate mini sessions. Unfortunately, unlike the total session duration cutoff of 30 minutes, there is no widespread threshold for page-stay time (Spiliopoulou et al. 2003). Therefore, we used a set of time thresholds (1, 2, and 5 minutes) in the test.

Table 4. Path Characteristics of Mini Sessions

Number of Paths	Number of Mini Sessions		
	t=1 min	t=2 mins	t=5 mins
2	27,140	23,485	20,964
3	4,457	4,242	4,075
4	1,340	1,469	1,427
5	477	590	652
6	212	268	285
7–10	183	230	240
>11	3	8	7
Total	33,812	30,292	27,650

Srikant and Yang (2001) describe an algorithm to discover the backtracking pages caused by clicking "back" button in the browser. We modified the algorithm such that it also identifies the backtracking pages where users revisit an already revisited page by other means, e.g., clicking a link to return to the homepage. Table 4 shows the number of mini sessions consisting of a given number of paths (>1) for different time thresholds applied on the data. Table 5 shows the number of mini sessions consisting of a given number of pages as we vary threshold values. As expected, the number of mini sessions obtained from the log data decreases as the time threshold value increases.

Table 5. Page Characteristics of Mini Sessions

Number of Pages	Number of Mini Sessions		
	t=1 min	t=2 mins	t=5 mins
1	308,261	222,734	172,248
2	87,174	62,594	49,297
3	45,459	35,662	28,893
4	28,986	23,828	19,240
5	21,257	18,319	15,384
6–9	45,607	43,389	38,973
>10	25,648	31,587	33,991
Total	562,392	438,113	358,026

The number of mini sessions having at least two paths is small compared to the number of all mini sessions. This may indicate that this Web site provides a good navigation structure to its visitors, making it easy to find the target information. On the other hand, an examination of the user sessions reveals that a large portion of mini sessions contain very few pages. Table 5 shows that more than half of the mini sessions consist of only one page. There are four possible reasons to account for this phenomenon. First, because Web cache is widely deployed in order to reduce bandwidth usage and server load, subsequent requests may be satisfied from the cache so that they are not logged by the server. Gupta et al. (2007) provide a discussion on how browsers caches and multilevel caches (e.g., intermediate caches at proxy servers) may affect the use of Web log files to learn users' traversal information. Second, it is revealed that many requests are generated by users clicking on links returned from search engines like Yahoo!. These requests are not initiated by users whose purpose is to explore the Web site for specific information. Because search engines return pages pertaining to the targeted information, users may directly click on the targeted page without undergoing an exploration process. Thus, very few pages may be browsed if a user is directed from a search engine. Third, many users bookmark their favorite links for quick access. The use of bookmark menu allows users to revisit their desired information in one click instead of searching from the home page of the Web site. Finally, since we are only interested in the visitors whose purpose is to look for specific information, we filtered the visitors whose intention was not to hunt for a particular page.

Test on Data

We apply the Web site maintenance model with two values as the out-degree threshold C. We vary the path tolerance value (b) and the multiplier for the penalty term (m) to see how the results vary with respect to these values. The test results are presented in table 6. The integer programs were coded in AMPL and solved using CPLEX/AMPL 8.1 on a PC running Windows XP on an Intel Dual Core 2 6300 processor. The times for generating optimal solutions varied from 0.109 seconds to 0.594 seconds, indicating that our approach is practical for real Web sites. The solution times are surprising given that we are solving an NP-complete problem. Note that we have reported the times taken to solve the integer programs only (the times taken for preprocessing steps like obtaining a_{ijkm}^{S} are not included as this can be done very effectively in practice).

Table 6. Results From the Log File Data

			The out-degree threshold (C)=20				The out-degree threshold (C)=40			
Time to break mini sessions (t)	Multiplier of penalty term (m)	Path Tolerance (b)	Number of new links	Number of links to be improved	Total number of excessive links	Time (sec)	Number of new links	Number of links to be improved	Total number of excessive links	Time (sec)
1 min	0	1	9,180	1,960	7,325	0.109	9,180	1,960	4,126	0.11
		2	1,643	1,698	3,666	0.438	1,643	1,698	1,292	0.422
		3	590	1,036	2,850	0.391	590	1,036	874	0.391
	1	1	9,180	1,960	9,717	0.125	9,190	1,960	3,804	0.125
		2	1,661	1,698	3,345	0.500	1,714	1,698	827	0.5
		3	616	1,036	2,596	0.406	621	1,036	678	0.375
	5	1	9,180	1,960	9,717	0.125	9,213	1,960	3,790	0.125
		2	1,681	1,698	3,333	0.516	1,932	1,698	728	0.578
		3	647	1,036	2,577	0.406	681	1,036	652	0.375
2 mins	0	1	7,895	1,871	8,722	0.125	7,895	1,871	3,790	0.109
		2	1,523	1,792	3,589	0.407	1,523	1,792	1,342	0.422
		3	633	1,198	2,876	0.359	633	1,198	908	0.375
	1	1	7,897	1,871	8,651	0.125	7,909	1,871	3,451	0.125
		2	1,541	1,792	3,261	0.484	1,609	1,792	840	0.485
		3	658	1,198	2,611	0.391	671	1,198	686	0.375
	5	1	7,897	1,871	8,651	0.125	7,937	1,871	3,436	0.125
		2	1,563	1,792	3,250	0.500	1,839	1,792	736	0.594
		3	695	1,198	2,590	0.485	744	1,198	652	0.391
5 mins	0	1	7,158	1,813	8,123	0.125	7,158	1,813	3,557	0.125
		2	1,375	1,825	3,504	0.39	1,375	1,825	1,325	0.391
		3	562	1,236	2,848	0.36	562	1,236	922	0.375
	1	1	7,158	1,813	8,049	0.125	7,180	1,813	3,209	0.141
		2	1,393	1,825	3,193	0.484	1,452	1,825	853	0.469
		3	585	1,236	2,609	0.407	609	1,236	688	0.375
	5	1	7,159	1,813	8,048	0.141	7,195	1,813	3,199	0.141
		2	1,409	1,825	3,182	0.484	1,706	1,825	738	0.5
		3	619	1,236	2,588	0.515	679	1,236	652	0.359

The column "Number of new links" indicates how many new links need to be inserted into the current Web site so that the visitors are able to locate the target information in at most b paths, which is specified in "Path Tolerance (b)" column. For example, if we use 2 minutes as the time threshold value and do not penalize on pages having excessive number of links, i.e., m=0, then we need to add 7,895 new links to the current Web site in order to make sure that the visitors are able to locate their targets in only one path with no backtracking. This translates to about 8.62 new links needed per page. The number of new links that should be inserted is very small considering the number of mini sessions that are required to be improved (30,292). In this example, approximately one link is required for every four mini sessions. Along with 7,895 new links, there are 1,871 existing links needed to be reported to Web designers for improvements, which translates to about two links per page.

The table shows that the numbers of existing links that are to be improved are the same no matter what value of C is used. This is because we use $(1-\lambda_{ij})$ in the objective function so that the existing links are selected whenever possible. We note that all mini sessions obtained from the log data were used in our test. In practice, we may use machine learning techniques to identify the important groups of mini sessions from, for example, users' demographic information, and apply our model only on these mini sessions.

The number of new links to be added drops significantly to 1,523, if a higher path tolerance value is selected, say, b=2 (the other parameters remain unchanged). This is partly because when a greater b is used, the mini sessions whose number of paths is lower than b are excluded from the optimization, since they already satisfy the imposed minimum satisfaction constraint. Therefore, when set b=2, the mini sessions consisting of only 2 paths (23,485 mini sessions) already satisfy the requirement and are not taken into account in the IP model. Only 6,807 (30,292−23,485) mini sessions are considered when b=2 as compared to 30,292 mini sessions when b=1. The other reason is that when a greater b is used, the number of potential links per mini session also increases. So, the optimization space is greater when a bigger b is chosen. The optimization results show that only 1 new link is needed per 4.5 mini sessions and only 1.66 new links are required per page. The number of new links that are required for b=2 is far fewer than for b=1. In general, the higher the value of b, the fewer number of new links that are to be introduced. We note that only one value of b is used for all pages in our test. In practice, the Webmaster should choose appropriate b values for different Web pages depending on their importance, as discussed in the previous section.

The result shows that when the out-degree threshold C=20 and the path tolerance value b=1, the number of new links is not sensitive to the values of multiplier of penalty term (m). For example, when using 5 minutes as the time threshold (t=5mins) and setting C=20 and b=1, the number of new links are 7,158, 7,158 and 7,159 for m=0, 1 and 5 respectively. It seems that the value of m does not affect the number of new links significantly when b and C are small. However, when a greater value of b is used or we increase the out-degree threshold value, e.g., set C=40, the changes in number of new links are more visible when we vary the values of m. For example, when setting t=2mins, and b=3, the number of new links are 633 and 695 for m=0 and m=5 respectively. The result indicates that 62 more new links are to be added when we associate a high value to the multiplier of the penalty term in the objective function. Although more new links are to be inserted to the current structure when m=5, we have 286 (2,876–2,590) fewer links to be added to the nodes whose out-degrees already exceed the threshold value, i.e., C=20. This can be interpreted as when a high penalty is imposed to pages with many links (in our case, m=5 and C=20), the optimization model will choose to add more new links to pages with smaller out-degree instead of adding fewer links to pages who already have many links. We also note that when b=1, even the number of new links does not vary much when we change the value of m, the total number of links exceeding the out-degree C is smaller when a higher value is used for m. For example, when C=20 t=5, b=1, the number of new links does not change when we increase m=0 to m=1. Nevertheless, the total number of links that are there in pages whose out-degree exceeds the out-degree threshold declines from 8,123 to 8,049.

The reasons for the above phenomenon are as follows. When b=1, only the nodes that appear in the first path of the obtained mini sessions are considered to link the target, so the number of potential links for each mini session are relatively small, with many having 1 or 2 such links only. When C is relatively small, many of the candidate nodes ($i \in N_i$, $N_i = \{i: (i, j) \in E\}$) for linking to the target nodes ($j \in N_j$, $N_j = \{j: (i, j) \in E\}$) already have out-degrees greater than the value of C, and few candidate nodes have out-degrees less than the threshold values. Therefore, when a small value is set to both b and C, it is possible that many nodes whose out-degrees already exceed threshold value must be selected to link to the target in order to satisfy the constraints. For these nodes, whether having a penalty term in the objective function does not change the results, as no matter which value m takes, they have to link to targets anyway and their values are already greater than the threshold value. In this case, the penalty term in the objective function does not affect the optimization result much. When the value of b increases, the number of potential links for each mini session also increases and so does the number of choices per mini session. Similarly, when the value of C becomes greater, there are more candidate nodes with an out-degree less than C and fewer nodes with an out-degrees greater than C. Therefore, our model will choose to add more links to the pages whose out-degrees are less than C instead of adding links to nodes whose out-degree already exceeds C as m increases. This is also the reason that when m=1 and 5, and the same value of b is used, the number of new links is greater for the case when C=40 compared to when C=20. For example, when t=2, m=5, b=2, the number of new links to be added are 1,563 and 1,839 for C=20 and C=40 respectively.

Discussion

Dynamic Web sites

Unlike static Web sites whose structures are carefully designed by Web designers, dynamic Web sites normally do not have a pre-designed structure, because the dynamic pages are generated as a result of queries run on underlying database. Dynamic pages are generally used in service-oriented Web sites to allow users to complete tasks through interacting with an online application. Static pages are more often seen in Web sites whose purpose is to facilitate users to quickly find their information goal through navigation, e.g., informational Web sites. In this sense, as stated in the introduction, our approach is primarily applicable to static Web sites. Dynamic Web pages can pose difficulties on methods that focus on improving Web site structure. First, dynamic Web sites, unlike static Web sites, do not have a former structure, because a page is generated each time a user requests it, rather than physically stored in the hard disk. Second, Web sites like amazon.com append a unique session id to every dynamically generated URL. Thus, a dynamic Web site can have an infinite number of pages, making it infeasible to optimize every possible page. Last but no least, since dynamic Web pages are generated "on the fly", it is compulsory to optimize the link structure in real-time, resulting in a very high workload on the server side.

Having stated these difficulties, our proposed approach may still be applied on dynamic pages depending on how they are generated. If templates are used to generate dynamic Web pages, our approach can be used to improve the set of predefined links in the templates, because only the content of the page changes, not the links embedded in the page. If the personalization technology, instead of a set of templates, is used such that the organization of a dynamic page largely depends on, for example, the user profile and the set of visited pages of the user, our approach may not be applicable. Such a case belongs to the study of personalization algorithms that can optimally generate Web pages based on user preferences and browsing history, not within the scope of our work.

Heterogeneous Web Site Users

In this paper, we described a special case where there is only one site structure available for all users. When there are several types of users, and technology allows the Web site to properly identify the user types, then several site structures may be made available such that users with different navigation preferences can choose the appropriate site structure. The combined hierarchy used in (Fang and Holsapple 2007) is an example which allows a user to select one of the available structures for navigation. Accordingly, our approach can be used to optimize the site structures for different types of users based on their log file data. In fact, dynamic Web sites that employ personalization techniques are an extreme case in which each user is considered a unique type and presented with a personalized Web site structure which, in theory, perfectly suits that user. However, this belongs to the field of personalization approaches, not the domain of transformation approaches.

Limitations of Experiments and Directions for Future Research

Though the experiment result in the previous section is very promising, there are several limitations in our experiment. Indeed, these are areas in which further investigation is needed.

In the paper, we presented the test result of our integer programming model on one Web site. This site comprises more than 900 Web pages and indeed has a reasonably sophisticated structure, as Adamic and Huberman (2001) find that the vast majority of the Web sites have fewer than 1,000 pages. Having said this, we also believe that further experiments need to be performed in order to generalize our result to the other types of Web sites whose characteristics are different from the one we tested. In addition, these experiments will help gain more insight on the parameter selection for the log preprocessing and the integer programming model.

The proposed model is shown to perform very efficiently and generate satisfactory results. Nonetheless, we have not empirically verified the improved Web site. There are two possible ways to conduct such an empirical validation. One way is to collect the user behavior data from both the original Web site and the improved Web site, and then evaluate their actual usability. Such a comparison can provide a strong support on whether the improved Web site indeed facilitates effective user navigation. The other way is to partition the available data set into two parts: one for training and the other for testing. The training data is used by the integer programming model to generate the structure of the improved Web site. Subsequently, the testing data is applied on the improved structure to

approximate the actual use of the improved Web site. In this way, the user behavior is simulated on the improved Web site and the result can be compared against that from the original structure.

Conclusions

We have proposed an integer programming model to mitigate the discrepancy between the Web site structure and visitors' expectation while preserving the current structure by minimizing unnecessary changes to a site. Our approach is designed to improve a Web site on a regular and progressive basis and therefore can be applied for Web site maintenance. The tests on a data set taken from a real Web site show that our approach can provide significant improvements over the structure of the Web site by introducing only a small number of new links. In addition, optimal solutions are usually obtained in a fraction of second, suggesting that our model will be very effective even to large Web sites in real word.

The performance of our method, like other related works, depends on the accuracy of techniques to demarcate mini sessions from Web log files. The time threshold used will have effects on which pages are selected as target pages. In this sense, techniques that can accurately identify target pages for a visitor are very important to our method. Therefore, futures studies may focus on developing techniques that can accurately identify users' target pages.

Weinreich et al. (2008) conduct a long-term client-side Web usage study and present new empirical findings on the changes on user browsing behaviors. How these new user navigation behaviors affect the model of this paper deserves further investigation.

References

Adamic, L. A., and Huberman, B.A. "The Web's hidden order," *Communications of the ACM* (44:9), 2001, pp. 55–59.

Chen, M.S., Park, J. S., and Yu, P. S. "Efficient Data Mining for Path Traversal Patterns," *IEEE Transaction on Knowledge and Data Engineering* (10:2), 1998, pp. 209–221.

Chiger, S. "Benchmark 2006 on E-commerce," Multichannel Merchant, 2006. (available at http://multichannelmerchant.com/webchannel/ecommerce_benchmark_05012006/index.html).

Cooley, R., Mobasher, B., and Srivastava, J. "Data preparation for mining World Wide Web browsing patterns," *Journal of Knowledge and Information Systems* (1), 1999, pp. 1–27.

Dhyani, D., Ng, W. K., and Bhowmick, S. S. "A survey of Web metrics," *ACM Computing Surveys* (34:4), 2002, pp. 469–503.

Fang, X., and Holsapple, C. "An empirical study of web site navigation structures' impacts on web site usability," *Decision Support Systems* (43:2), 2007, pp. 476–491.

Fu, Y., Shih, M.Y., Creado, M., and Ju, C. "Reorganizing Web sites based on user access patterns," *Intelligent Systems in Accounting, Finance and Management* (11:1), 2002, pp. 39–53.

Garey, M., and Johnson, D. *Computers and Intractability: A Guide to the Theory of NP-Completeness,* San Francisco, CA: W. H. Freeman, 1979.

Greenspan, B. "Web site spending picks up," The ClickZ Network, 2004. (available at http://www.clickz.com/showPage.html?page=3356871).

Gupta, R., Bagchi, A., and Sarkar, S. "Improving linkage of Web pages," *INFORMS Journal on Computing* (19:1), 2007, pp. 127–136.

ISO 9241-11. *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: guidance on usability*, Geneva, ISO, 1998.

Lazar, J. *Web usability: a user-centered design approach*, Boston, MA: Pearson Addison-Wesley, 2006.

Lin, C. C. "Optimal Web site reorganization considering information overload and search depth," *European Journal of Operational Research* (173), 2006, pp. 839–848.

Lin, W., Alvarez, S., and Ruiz, C. "Efficient adaptive-support association rule mining for recommender systems" *Data Mining and Knowledge Discovery* (6), 2002, pp. 83–105.

Marsico, M.D., and Levialdi, S. "Evaluating Web sites: exploiting user's expectations," *International Journal of Human-Computer Studies* (60:3), 2004, pp. 381–416.

Nakayama, T., Kato, H., and Yamane, Y. "Discovering the gap between Web site designers' expectations and users' behavior," *Computer Networks* (33), 2000, pp. 811–822.

Otter, M., and Johnson, H. "Lost in hyperspace: metrics and mental models," *Interacting with Computers* (13), 2000, pp. 1–40.

Palmer, J. "Designing for Web site usability," *IEEE Computer* (35:7), 2002, pp. 102–103.

Perkowitz M., and Etzioni, O. "Towards adaptive web sites: Conceptual framework and case study," *Artificial Intelligence* (118), 2000, pp. 245–275.

Pirolli, P., and Card, S. K. "Information Foraging," *Psychological Review* (106:4), 1999, pp. 643–675.

Spiliopoulou, M., Mobasher, B., Berendt, B., and Nakagawa, M. "A framework for the evaluation of session reconstruction heuristics in Web-usage analysis," *INFORMS Journal on Computing* (15:2), 2003, pp. 171–190.

Srikant, R., and Yang, Y. "Mining Web logs to improve Web site organization," in *Proceedings of the 10th International conference on World Wide Web*, Hong Kong, 2001, pp. 430–437.

Weinreich, H., Obendorf, H., Herder, E., and Mayer, M. "Not quite the average: an empirical study of Web use," *ACM Transactions on the Web* (2:1), 2008, pp. 5:1–5:31.

Yen, B.P.-C. "The design and evaluation of accessibility on web navigation," *Decision Support Systems* (42:4), 2007, pp. 2219–2235.