Design and Evaluation of Improvement method on the Web Information Navigation - A Stochastic Search Approach

Benjamin P.-C. Yen
School of Business
The University of Hong Kong, Hong Kong
benyen@business.hku.hk

Y.-W. Wan
Department of IEEM
HKUST, Clear Water Bay, Hong Kong
ieywan@ust.hk

Abstract

With the advent of fast growing Internet and World Wide Web (WWW), more and more companies start the electronic commerce to enhance the business competitiveness. On the other hand, more and more people surf on the Web for information gathering/processing. Due to unbalanced traffic and poorly organized information, users suffer the slow communication and disordered information organization. The information provider can analyze the traffic and Uniform Resource Locator (URL) counters to adjust the organization; however, heterogeneous navigation patterns and dynamic fluctuating Web traffic make the tuning process very complicated. Alternatively the user may be provided with guidance to navigate through the Web pages efficiently. In this paper, a Web site was modeled as a Markov chain associated with the corresponding dynamic traffic and designated information pages. We consider four models: inexperienced surfers on guidance-less sites, experienced surfers on guidance-less sites, sites with the mean-length guidance, and sites with the known-first-arc guidance (generalized as sites with dynamic stochastic shortest path guidance). Simulation is conducted to evaluate the performance of the different types of navigation guidance. We also propose a reformulation policy to highlight the hyperlinks as steering guidance. The evolution on complexity and applicability is also discussed for the design guideline of general improvement methods. The paper concludes with the summary and future directions.

1. Introduction

Information plays an indispensable role in the real world. The network and information systems have changed the way people communicate with each other as well as expedited the process to obtain the information that matches their interests. Everyday, hundreds of millions of transactions flow through the network all over the world. Any information can be transferred from one extremity to the other on the earth only within few seconds. Together with the growth of the needs of information, the web pages on the Internet grow explosively during the past few years and such increase is expected to be more acute over time.

The rich URL's are no doubt a great wealth for all the visitors to retrieve the comprehensive information on the Web. The Web page owner can profit from the advertisement when visitors drop by the web pages. However, too much unorganized information may create problems with searching performance in the meanwhile. Visitors spend astounding amount of time in navigating through the useless or redundant pages. Web page owners need to increase the investment to update and reorganize information on Web pages on-line or periodically. As a result, visitors and web page owners are facing the same problems: How to balance the gain and the cost so that both can get the most satisfaction? How can the Web site serve the visitors better for information retrieve? Is it worthwhile for all the effort and money the web page owners need to invest? Some related research has been done for Web information modeling and performance analysis to improve the searching process and resource allocation.

The research for information access on WWW is mainly divided into three groups based on their application and scope – the web site customization based on the user access information, the search for information retrieval and discovery, and the intelligence browser to support user navigation and the collection of user information on the Web.

Perkowitz and Etzioni [1-2] propose an Artificial Intelligence approach to create the Adaptive Web Site, which can improve the site organization based on the users access log with the assumption of each originating computer corresponding to a particular user. Yan *et al.* [3] propose the use of access patterns to generate hyperlinks, which are captured in the access log and analyzed offline in an interval basis, to improve the information access. Wang *el al.* [4] propose a personalized filtering model to filter and rank the product information with linear functions on the user preference.

The research in the second group focuses on seeking for the information on the Internet. Cheung *et al.* [5] propose a model of four-level classification tool of learning the behavior of both information user and information source. Chen and Kuo [6] propose a

Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)
0-7695-1874-5/03 $17.00 © 2002 IEEE

personalized information retrieval system based on the user profile modeled as the Semantic Relevance (SR) and Co-occurrence (CO) of keywords to capture the real meaning of user query. Chang *el al.* [7] present a Site Traveling Algorithm (STA) to discover the relevant information, in which the relevance of the retrieved document is evaluated with the content popularity and richness (CPR). Yang *el al.* [8] present the development of intelligent personal Internet agent based on automatic textual analysis of Internet document and hybrid simulated annealing algorithm. Tu and Hsiang [9] propose an Interactive Information Retrieval (IIR) agent architecture to handles group knowledge and preference, and to keeps track of the individual user profile. Teng *et al.* [10] propose a scalable method for parallel processing in both information crawling/gathering and processing.

The third group of research concerns navigation assistant for user during the browsing process. Joachims *et al.* [11] introduce the Web-Watcher, based on a learning approach with the user feedback to improve the quality of advice for navigation interactively. Similarly, Liaberman [12] introduces the intelligent agent, *Letiza*, which works with conventional web browser to keep track of the user browsing behavior and interests. Furthermore, Berghel *et al.* [13] present a web browser called the "Cyberbrowser" to customize the information access for the content within the web page, which include keyword and sentences extraction according to user selection. Lin *et al.* [14] describe an approach for capturing user access patterns on the WWW to address the problem that the web server will only recognize the proxy server instead of the individual user. The method used is called "page conversion" and each page in the site is encoded into a cipher in the server-side. When user requests a page, a client-side program (deciphering module) is downloaded from the server and report the event of page access to the Access Pattern Collection Server (APCS) before decipher the encoded page and present to the user. Richardson [15] does a comparison on the existing tools to gather the access information on the Internet, such as visitor counter and guest book.

The literature above shows the modeling of the information retrieval mostly is based on database models and data mining techniques. The analysis centers on the user behavior patterns largely for the global Web information retrieval. There are few studies for the analysis on site structure on information retrieval as optimization problems. There are two types of models for problem formulation – deterministic models and stochastic models. Gibson *et al.* [16] and Chakrabarti *et al.* [17] define the Web sites as "authorities" and "hub" in isolation and conclude that a respected authority is page that is referred to by many good hubs and a useful hub is a location that points to many valuable authorities. Chakrabarti *et al.* [18] develop algorithms that exploit the hyperlink structure of the WWW for information discovery and categorization, the construction of high quality resource lists, and the analysis of on-line hyperlinked communities. Kleinberg *et al.* [19] describe two algorithms that operate on the Web graph, addressing problems from Web search and automatic community discovery.

Sarukkai [20] uses a Markov Chain model based on the user access information for link prediction and path analysis. Levene *et al.* [21] derive Zipf's rank frequency law from an absorbing Markov chain model of surfers' behaviour assuming that less probable navigation trails are, on average, longer than more probable ones. Levene and Loizou [22,23] formulate a hypertext database as a graph and propose a probability approach to find the trail to match the query. Kumar *et al.* [24] propose a stochastic model of the web graph to show some addition properties for the random graph. Levene *et al.* [25] extend the evolutionary model of the Web graph by including a non-preferential component and viewing the stochastic process in terms of an urn transfer model. By making this extension we can now explain a wider variety of empirically discovered power-law distributions provided the exponent is greater than two. Zin and Levene [26] propose that information on the topology is important for useful exploration and can also help to reduce the feeling of disorientation that users experience

From the review above, it should be noted that there lacks the consideration of some dynamic characteristics (such as dependent reverse links and user familiarity with Web navigation) and evaluation on the access models. In this research, we take into account the dynamic characteristics for problem analysis and we propose various dynamic policies with performance evaluation. We propose a graph-based structure with stochastic property (such as traffic conditions and navigation information) and various dynamic policies to guide users in information access. The Web structure and routing policy have been addressed based on discrete-time Markov chains, and the expected time for a surfer to get to his destination is found from dynamic and stochastic shortest paths. We consider four models: inexperienced surfers on guidance-less sites, experienced surfers on guidance-less sites, sites with the mean-length guidance, and sites with the known-first-arc guidance (generalized as sites with dynamic stochastic shortest path guidance). We also conduct numerical experiments and simulation to evaluate the access performance for these models. Section 2 describes the problem description and formulation. Four searching models for navigation guidance are discussed in section 3. We demonstrate the simulation result for performance evaluation of these models in section 4. We also propose a reformulation policy and the evaluation of four models from the aspects of complexity and

applicability. The paper concludes with the research summary and future directions.

2. Preliminary – problem formulation

A Web site consists of a number of Web pages, and each page may have hyperlinks connecting to other pages. Each page is associated with a download time, which varies with respect to the page content and the network conditions. Vertices and arcs are denoted as follows:

Vertices: $V=\{v_1, v_2, \ldots v_n\}$. Each v_i (i=1, 2, … n) refers to one page. $N=\{z_1, z_2, \ldots z_m\} \subseteq V$ represents the set of destination vertices, which corresponds to the pages to be visited ultimately; z_1 is the root.

Arcs: $A=\{[v_i, v_j]$ or e_{ij} | There is a hyperlink in page i pointing to page $j\}$. The arc connecting v_i to v_j denotes a hyperlink from page i to page j.

Note that the hyperlink is directed, so is the network. Since a Web page has a download time, the corresponding vertex in the network is assigned a weight representing such time. Naturally, the download time of a Web page is directly determined by the size of its contents, such as text, images and sound/video clips, and network conditions. Among these factors, the page size (x_i) is the most important one. Hence, it is assigned as the weight of vertex to reflect the download time:

$$w: V \rightarrow R^+$$
$$w(v_i) = x_i$$

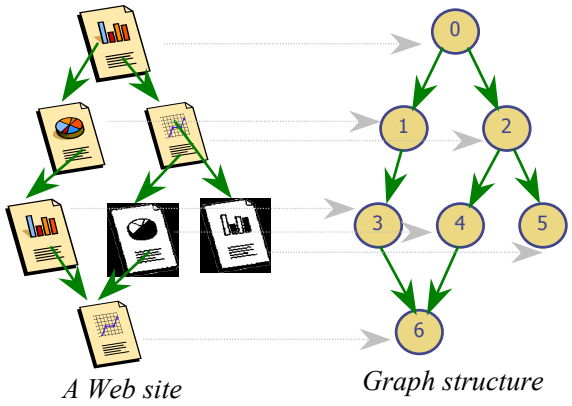

Figure 1. A Web site and its graph structure

Therefore, the network is denoted as $\vec{G} = (V, A, w)$, which is a *directed graph* in which a weight is associated with each vertex. Figure 1 shows an example for a Web site and its graph structure.

For example, given a web structure in Figure 2, there are several ways to reach the destination L from A. Suppose A-C-G-L is the shortest path. Sometimes due to the traffic jam, we may select different path (node by node) during the navigation. If we consider the cache function is enabled, the length (loading time) for each reverse link becomes zero; otherwise, the reverse link bears the loading time of the previous page. For a multi-destination problem, we need to find a shortest path to cover the all the destination nodes. If we allow modifying the hyperlinks in Web pages, then we may delete/add some links to improve the web structure.

2.1. Classification

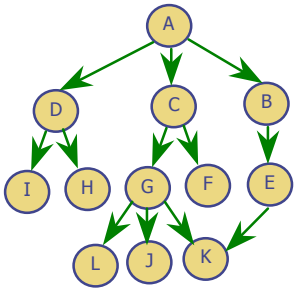

Figure 2. Example – a Web structure

We can extend the model for information retrieval on the Web into two directions – structure properties and navigation properties. The *structure properties* concern the static properties of the Web site structure, including the server capacity and the cache mechanism. The *navigation properties* are related to the dynamic aspects of the information retrieval, including the single or multiple root pages and destination pages, constraints on the navigation path, etc. Both types of properties have great impact of modeling and solution approaches as a searching problem on a general graph.

Structure properties. The server performance might be inversely proportional to the number of the users request the pages simultaneously. This can be valid for the individual page or the Web site as a whole. One special characteristic for the Web site structure is the "conditional reverse link" – the hyperlink (arc) visited enables the corresponding reverse link. The load time associated with the reverse links depends on the cache function, i.e. it is zero if the cache is on or the time to load the previous page otherwise. The cache can be used for both Web page address and page content. There can be some constraints on the size of cache. Furthermore, the links can be bi-directional (i.e. the links in both directions are valid in the original graph) and multiple (i.e. multiple identical or non-identical links between nodes).

Navigation properties. The navigation can be a single root page (entry page) or multiple root pages (direct page address). Similarly, the destination page might be a single page or a group of the pages. The group of pages can be contracted into a simplified structure by merging them and the inter-links between the nodes. The navigation can be constrained by path length or time for distraction or retreat. The objective of the navigation can be related to

time, information (completeness and relevancy), and path length (radical distance from the root and number of pages visited). The decision of the navigation can be decided as static or dynamic model. In the static model, the whole navigation path is generated in a batch; however, the dynamic model handles the next nodes to visit one by one.

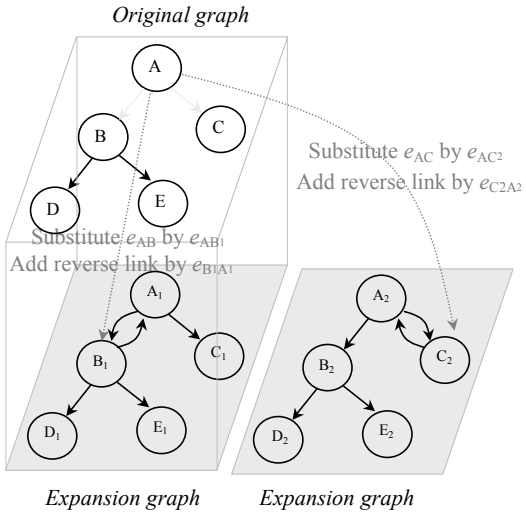

Figure 3. Example of explorative transformation of link e_{AB}

2.2. Problem Transformation

Based on the description of properties mentioned above, the model can be extended by taking into account more realistic consideration. The *explorative transformation* approach for conditional reverse link is discussed to demonstrate the model extension. We may start with the original Web graph without considering the conditional reverse link. We may substitute each link by a new link points to the corresponding nodes in a duplicated graph and add the reverse link in the new graph. In Figure 3, the link e_{AB} in the original graph is substituted with a new link e_{AB1} pointing to a duplicated graph (expansion graph). A new link e_{B1A1} is added to the graph to represent the "back" function. The loading times of e_{B1A1} and e_{A1B1} depend on the cache policy. We can continue the process similarly until all the links in the original graph are replaced and the corresponding expansion graphs are generated. Then we can again continue the similar process on the expansion graphs to further the link replacement and graph expansion. In the process of expansion, the inter-links between graphs are built-up until all the non-visit links are replaced. There are in total 2^m graphs after the expansion where m is the number of the links in the original graph. All the root and destination pages still serve the same role in the expansion graphs.

In this study, we focus on cases with the assumption of single-destination and disabled cache. We also demonstrate the reformulation policy to re-sequence the links for navigation guidance without changing adding or deleting the links in section 5.

3. Searching model

Surfers are of different skill levels and sites are of different degrees of maturity. An inexperienced surfer moves among and loads randomly pages of a site, while an experienced one checks pages systematically and repeats loading a page only if necessary. An immature site does not provide any navigation information for surfers, while a well-designed site will provide navigation guidance, with information on the site structure and (expected) loading times of pages. A sophisticated site may even provide real-time navigation information that changes with the traffic of the site.

In this section, we will model the page searching of surfers of different skill levels on sites of different degrees of maturity. We represent the loading times by page-dependent random variables. Such an approach is a first-order approximation that captures the variation of the loading time with the traffic within the site. For simplicity, we assume that the cache function is disabled and that the surfer only has one destination in mind. Sites with cache function enabled and multi-destination are left for future extension work. We consider four models: inexperienced surfers on guidance-less sites, experienced surfers on guidance-less sites, sites with the mean-length guidance, and sites with the known-first-arc guidance (generalized as *DSSP* guidance). We will illustrate the models with a site structure shown in Figure 2. The nodes are the pages and the arcs are the hyperlinks of a page. We take page *A* as the root and page *L* as the surfer's destination, and we will compare the expected time to get to page *L* after page *A* has been loaded. All loading times are assumed to be independent, while the page itself characterizes the distribution of the loading time of a page.

The modeling is mainly is based on the work of Glover *et al.* [27], Shier and Witzgall [28], Psaraftis and Tsitsiklis [29], Geetha and Nair [30], Polychronopoulos and Tsitsiklis [31] and Cheung [32]. Cheung [32] studies the formulation of a dynamic shortest path in a network and proposes a routing policy to compute the expected path cost by mimicking the classical label-correcting approach.

3.1. Inexperienced Surfers On Guidance-less Sites (IL)

First consider a totally inexperienced surfer on a site without any navigation guidance. The surfer moves randomly, possibly back and forth among the pages, and picks links in a page arbitrarily. The movement of such a surfer can be modeled by a random walk on a connected graph, and the page search process can be modeled as a discrete-time Markov chain [20].

Refer to the site structure in Figure 1. Let $X_n = s$ if the surfer is at page s after the nth move (page loading). Take $X_1 = A$, because A is the root page. In general, if a surfer starts from page s with probability p_s, we can take $P(X_1 = s) = p_s$. Let $S = \{A, B, \ldots, L\}$ be the state space of $\{X_n\}$. From the description on the above paragraph, $\{X_n\}$ is a discrete-time Markov chain. The transition probabilities can be found from the number of links on a page. For example, if the surfer is on page G of the site shown in Figure 2, he will next visit sites C, K, J, and L with probability 0.25 (providing that he decides to continue his surfing). It is straightforward to show that $\{X_n\}$ is an absorption chain with L as the absorbing state.

For any page $s \neq L$, let N_s be number of visits to page s before visiting page L and T_s be the loading time of page s. $E[N_s]$ is found from the first passage time argument from state A to state L for the chain $\{X_n\}$ (please see [33], pp 152 and pp. 172) and the expected search time of $L = \sum_{i \neq L} E[N_s]E[T_s]$.

If we further extend the model by considering the reverse links, then we may need to extend the graph in explorative transformation to ensure the directions of the link and the conditional activation of reverse links. We may use the simulation model based on the extended graph to calculate the expected search time.

3.2 Experienced Surfer On Guidance-less Sites (EL)

Now consider an experienced surfer visits for the first time a site that does not provide any tour guidance. The surfer randomly picks up *unvisited* links in page. As far as possible, he will not re-visit a page that he has visited before. However, he still needs to re-visit some pages when he goes into a dead end during searching for his destination.

Such a search behavior cannot be modeled by the random walk in Section 3.1. We can still formulate it as a discrete-time Markov chain, but the size of the state space will be astronomically big: for N pages, to keep track of the identification of the pages visited, the state space is of size $O(N2^N)$. The expected search time will be the first-passage time from first entering page A, to any state such that page L is visited for the first time. The size of the chain precludes any sensible study through this approach on sites of practical value. Fortunately, we can still easily build up a simulation model to estimate the expected search time for this approach.

We may also extend the model by taking into account of conditional reverse links. Similar to that in section 3.1, the model can be extended by explorative transformation that can be combined with the extension for discrete-time Markov chain mentioned above.

3.3. Sites with Mean-Path Guidance (MP)

From this section onwards we consider sites that provides various types of navigation guidance to surfers. The navigation guidance may be static or dynamic, ranging from long-term mean to exact loading times (of that moment), with all possible combinations of means and exact values lying between the two extremes. Because there is navigation guidance, the difference between the expected search times of experienced and inexperienced surfers is minimal and hence is ignored.

In this subsection, we consider tour map guidance

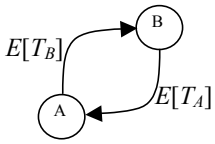

Figure 4. Example – Arc replacement

constructed from the mean path (loading) time. To do so, first replace each of the undirected arcs of a site structure by two directed arcs, one to each of the two nodes spanning a directed arc. The change of arc AB of Figure 4 is shown below. The *length* of arc $A \rightarrow B$ is $E(T_B)$, the expected time to load page B, and the *length* of arc $B \rightarrow A$ is $E(T_A)$, the time to load page A. Other arcs are treated similarly. After the forming the directed arcs with mean loading time, we get a directed graph with positive cycle

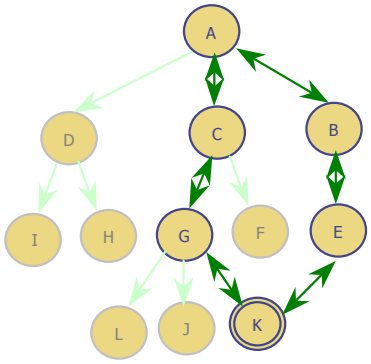

Figure 5. Cycling in DSSP Guidance

length. Standard algorithms, such as Dijkstra's algorithm

(or its variations) [34] can be used to find the shortest path from *A* to *L*.

In a site that provides the Mean-Path Guidance, the surfer is given the sequence of identification of pages found from the above procedure. While the calculation is simple and straightforward, the path from the Mean-Path guidance may not be fast if one knows the actual page loading times at that moment. For example, suppose that the Mean-Path guidance suggests the path *A*-*C*-*G*-*L*. However, at any epoch, due to the instantaneous traffic, the actual time taken for the path *A*-*B*-*E*-*K*-*G* could be less than that of *A*-*C*-*G*, which makes the path suggested by the mean path length not optimal.

We can further extend the model by taking into account the conditional reverse links. The explorative transformation inherits the lengths of the reverse links as mentioned above.

3.4 Sites with Dynamic Stochastic Shortest Path Guidance (DS)

The known-first-arc guidance is a simple extension of the mean-path guidance in which minimal real-time loading time information is considered. At page *A*, the surfer has three pages *B*, *C*, and *D* to choose from. If the system can estimate the loading time of pages *B*, *C*, and *D*, then in the calculation of the shortest path from *A* to *L*, the exactly loading times are used in arc *A*-*B*, *A*-*C*, and *A*-*D*, while the mean loading times are used in the other arcs. Such a method is called the *known-first-arc* guidance. It is simple to show that this guidance dominates the mean-path guidance, and the effect is more noticeable when loading times have large variance.

It is very inviting to apply the known-first-arc guidance repeatedly. Suppose that the surfer moves to page *C* based on the known-first-arc approach. Then at *C*, the loading times of pages *A*, *F*, and *G* become known quantities at the moment that the surfer leaves page *C*. The system can repeatedly apply the known-first-arc method to determine the next page to load, with the objective to move to page *L* in the shortest time. The guidance provides by this method is dynamic. Roughly speaking, after the surfer first loads root page and keys in his detention, the system will guide him through the site to the destination by informing him dynamically which page to load next.

Assume that each time the known loading times are random draws of the corresponding T_s. The guidance that we mention above is exactly the *DSSP* problem considered in Cheung [32]. See Cheung for algorithms to compute the expected shortest path in *DSSP*.

As discussed in Cheung, the *DSSP* provides means for the surfer to retreat from a wrongly chosen path, when actual rather than mean of some loading times are revealed. However, the surfer may cycle around pages before he reaches the destination. Refer to Figure 5 below that shows part of the site structure in Figure 2. For simplicity, we use arcs with two arrows to represent the two directional flows. Here we assume that the loading times of all pages are *i.i.d*. random variables, distributing uniformly in [1, 2, 3, 10], no matter a page is visited for the first time or not.

Suppose that the surfer wants to go from page *A* to *G*. At the moment that the surfer leaves *A*, going next to page *B* or *C* depends on the then current loading times of page *B* and page *C*. If loading time of page *B* is of one unit and page *C* of 10 units, then next loading page *B* is better; else loading page *C*. If the next loaded page is *C*, then the surfer is directed to page *G* next. However, if the next loaded page is *B*, it is possible for the surfer to be directed to page *A* or page *E*. Similarly, at page *E*, the surfer may be directed next to page *B* or page *K*.

The movement of a surfer based on the *DSSP* guidance can be modeled as a discrete-time Markov chain. Provided that there are not many links from one page to another, the first-passage analysis of such a chain is feasible for pages of reasonable sizes. Consider the same example in Figure 1 of going from page *A* to page *L*. Define a Markov chain $\{X_n\}$ with the same state and state space as in Section 3.1. The transition probabilities are found from the *DSSP* guidance. Let $S(i)$ be the set of successor pages possible to be visited next when a surfer is at page *i*. It is straightforward to find $p_{ij} = P(X_{n+1} = j|X_n = i)$ and $E[T_j|\{X_n\}$ moves from i to $j]$ from the joint distribution of $\{T_j, j \in S(i)\}$. Hence, the time from *A* to *L* is given by $\sum_{k=1}^{N_{AL}} T_k$, where N_{AL} is the first-passage time (number of transitions) from page *A* to page *L*, and T_k is the loading time taken in the *k*th page loading. N_{AL} can be expressed as the sum of the $i \to j$ transitions before reaching page *L*, and giving the $i \to j$ transition, the expected time is given by the set of conditional expected times $\{E[T_j|\{X_n\}\})$. Consequently, we can compute the expected search time $E\left[\sum_{k=1}^{N_{AL}} T_k\right]$.

We can further extend the model by taking into account the conditional reverse links. The reverse links can be dynamically added to the graph during the process of traversal. .

4. Numerical examples

In this section, we compare the expected search time of the five models discussed in Section 3. While some of these methods can be analyzed analytically, we use simulation to all five methods, a handy approach that is applicable to any complex site structure. For simplicity, we simulate on a site with a structure as shown in Figure

2, where page *A* is taken to be the root page, and page *L* as the destination of a generic surfer. All the page loading times are assumed to be *i.i.d*. random variables distributing uniformly in [1, 2, 3, 10]. With this choice of loading time distribution, the Mean-Path Guidance will definitely direct a surfer to the path *A*-*C*-*G*-*L*. The Known-First-Arc Guidance will direct a surfer either to path *A*-*B*-*E*-*K*-*G*-*L* or path *A*-*C*-*G*-*L*, depending on the instantaneous loading times of pages *B* and *C* when the surfer finishes with page *A*. The *DSSP* Guidance gives the same instruction as the Known-First-Arc Guidance when leaving *A*. However, at pages *B* and *E*, and any subsequent visits of page *A*, the page next visited depends on the real-time loading time information as discussed in Section 3.4.

We have simulated this simple site structure for 1,000,000 repetitions. Because the structure is simple, it is not necessary for us to use any sophisticated variance reduction methods. The simulation results are shown in Table 1 below.

Clearly our results are confounded by the scope, such as the structure of the site and the choice of loading time distributions, of our study. While we cannot take the specific numbers as concrete truth, we can still observe the general trend through these results.

Table 1. Simulation Results

Models	Mean Search Times[*1]
Inexperienced surfer on guidance-less Sites (IL)	161.50
Experienced surfer on guidance-less Sites (EL)	73.47
Mean-Path Guidance (MP)	12.01
DSSP guidance (DS)	11.95

[*1] The loading time of page *A* is excluded.

As expected, an inexperienced surfer without navigation guidance takes the longest time to get to his destination. Our results show a ratio of more than 13 times between this random search and the guided searches. An experienced surfer without guidance can cut down his search time by half than 50% when compared to an inexperienced surfer. He jumps back to a visited page only if he has exhausted all the possible options in a current page. However, without the knowledge of the structure of the site, repeatedly visiting a page is unavoidable. Consequently, its mean search time is more than six times of the guided ones.

The two guided searches give very close mean search times. The Mean-Path guidance is the worse than *DSSP* guidance among the three, because it does not make use of any real time information.

5. Comparison and evaluation

We have shown in Section 3 that any navigation guidance can significantly reduce the search time of surfers. The guidance can be implemented in (at least) two forms. In the first form, an explicit navigation map is given and the correct path is highlighted for surfers to follow. In the second form, sites structure can be dynamically re-arranged to suit the needs of surfers. The implementation of the first form is straightforward and its detail is skipped. Here we will discuss implementation issues of the second form.

5.1. Reformation approach

Refer to Figure 2 for the structure of the site for illustration. Remember that page *A* is the root and page *L* is the destination. The structure indicates the hierarchical arrangement of information: Page *A* contains links to pages *B*, *C*, and *D*, in the order of the links arranged in page *A*; other pages and links are interpreted in the same fashion. Given the known destination page *L* provided by the surfer, the system provides navigation guidance based on the real-time information at the moment that the surfer finishes a page. Suppose that the guidance directs the surfer to page *C*, and by the same token, next to page *G*. So that for the surfer, the site structure is as if that in Figure 6, all the relevant links being arranged in the most convenient top (right) position for the surfer.

In Section 4, we use simulation to evaluate the performance of the different guidance methods. In real-life, it is hard to simulate on real time for each surfer to determine his best virtual site structure. Fortunately, there are polynomial time algorithms to calculate the shortest paths. The standard shortest path algorithms for deterministic arc lengths give the shortest path for the Mean-Path guidance, as long as we set the arc lengths to their mean values. There is a family of similar algorithms to determine the shortest paths with real-time information on path length. On the following, we give the algorithm suggested in Cheung [32].

Any site structure is an undirected graph as shown in Figure 2. Let (G, N) be such a graph, where G = {*A*, …, *L*} be the set of nodes and N be the set of arcs of the graph. Node *L* is the destination. Let

$S(i)$ be the set of successor nodes of node i;

$B(j)$ be the set of predecessor nodes of node j;

T_{ij} be the (random) cost of arc (i, j); (T_{ij} is the loading time of page j in our application);

$\overline{V}_i$ be the expected distance from i to L.

As stated above, all the arc costs are assumed to be independent. Then

$$\overline{V}_L = 0,$$

$$\overline{V}_i = E[\min_{j \in S(i)} (T_{ij} + \overline{V}_j)], \ \forall i = A, \ldots, K. \quad (1)$$

Note that T_{ij}'s are known quantities whenever we compute the expected for node i. $\{\overline{V}_i\}$ are found by solving the equation set simultaneously, which can only be done numerically. Instead, Cheung suggested the following modified generic Label-correction method. It is an approximation in alternate to what we give in Section 3.4. Let

Q be the set of nodes (typically in the form of queue) whose distance from node L have been known;

$\hat{V}_i$ be an estimate of $\overline{V}_i$;

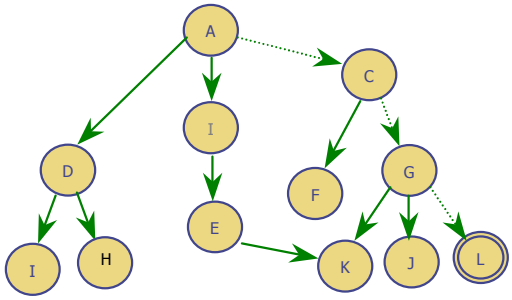

Figure 6. The Virtual Site Structure Experienced by a Surfer

$V_i^{\max}$ = maximum value of $\min_{j \in S(i)} (T_{ij} + \hat{V}_j)$.

Step 1 Initialize $\hat{V}_i = \infty, \forall\, i \in \mathrm{G}, i \neq L;\ \hat{V}_L$.

Step 2 Initialize Q.

Step 3 Remove a node i from Q and compute $\hat{V}_i = E[\min_{j \in S(i)} (T_{ij} + \hat{V}_j)]$.

Step 4 For each node $j \in B(i)$, if $V_j^{\max} < T_{ji} + \hat{V}_i$ and if $i \notin Q$, then add i to Q.

Step 5 Repeat steps 3 to 4 until Q = ϕ.

The ways to initialize Q (in Step 2) and to add i to Q are implementation specific. See Cheung [32] for an implementation example that reduces computational effort.

5.2. Evaluation and extension

All the four models work similarly as navigation guidance for information retrieval; however, they differ from the aspects of applicability and complexity. The *complexity* involves the performance of the model in the various scenarios, such as conditional reverse links, motivation guidance, decision-making, algorithm complexity, and structure modification. On the other hand, the *applicability* concerns the capability of problems types, such as cache option, multiple root nodes, multiple destination nodes, and concurrency. The following is the discussion of the comparison for four accessibility models from these two aspects.

A. Complexity

(1) *Conditional reverse links*. The original graph needs to be extended by explorative transformation for the models IL, EL and MP; however, it only adds the reverse links dynamically to the graph for model DS.

(2) *Navigation guidance*. The first two models, IL and EL, do not have the guidance, whereas the other two, MP and DS, benefit from the guidance of site structure and loading time for navigation decision.

(3) *Decision-making*. Both IL and MP models focus on the static decision-making, i.e. one-pass decision process. On the other hand, EL and DS models concern dynamic information for decision-making.

(4) *Algorithm complexity*. For IL model, the main task is to compute the first passage time which complexity is $O(N^2)$ for each node and total complexity is $O(N^3)$. Since the state space of the Markov chain for EL model, the total complexity is $O(N^4 2^N)$. The complexity of MP model depends on that for Dijkstra's algorithm, which is $O(N^2)$. The DSSP cannot be solved in polynomial time unless it is acyclic. If we take into account the conditional reverse links, the complexity of all four models are non-polynomial.

(5) *Structure modification*. The structure modification involves addition or modification links/nodes. For non-guidance models. IL and EL, are not effected by the changes for structure formulation; guidance-based models, MP and DS, need to update the guidance information. On the other hand, Static models, IL and MP, need to re-compute the solution by taking into account the new information; but dynamic models, EL and DS, consider the new information only during every dynamic decision process.

B. Applicability

(1) *Cache option*. The cache option decides the loading time for the conditional reverse links, which range between 0 (cache on) to the loading time of the original node (cache off). There may be some time (life span) constraints on the cache content. The cache option (either on or off) can be applicable for all four models. However, it is only applicable to SD model if we also consider the time constraint on the cache content.

(2) *Multiple root nodes*. The navigation may start from different root nodes in different sessions. All four models are capable of handling multiple toot nodes without any further modification.

(3) *Multiple destination nodes*. Unlike the case of multiple root nodes, the multiple destinations need to be covered in the same session. In the IL model, the

navigation can be counted as independent for the known-first-arc guidance, and *DSSP* guidance. From the

Table 2. Summary of comparison for searching models

	IL	EL	MP	DS
Complexity				
Conditional reverse links	Extend graph	Extend graph	Extend graph	Add links
Navigation guidance	No guidance	No guidance	Guidance	Guidance
Decision-making	Static	Dynamic	Static	Dynamic
Algorithm complexity *[1]	Polynomial	Non-polynomial	Polynomial	Non-polynomial
Structure modification	Re-compute	No changes	Update/ Re-compute	Update
Applicability				
Cache option *[2]	Not applicable	Not applicable	Not applicable	Applicable
Multiple roots	Applicable	Applicable	Applicable	Applicable
Multiple destinations	Linear	Non-Linear	Linear	Non-linear
Concurrency	Independent	Independent	Independent	Dependent

*[1] If taking into account conditional reverse links, the complexity becomes non-polynomial for all four models.
*[2] In the case of time-constraint cache option

destination nodes and the calculation is additive as liner function. Similarly, the MP model can also take the destination nodes as independent in the graph and calculate the result as a liner function. On the contrary, the calculation become much more complicated for both EL and DS models, which might take non-additive calculation as non-linear functions.

(4) *Concurrency*. The concurrency involves both multiple sources and destinations for different session (for different users) simultaneously. Since DS model needs real-time information for step-by-step decision-making, the multiple sessions might interfere with each other. For the other models (IL, EL, and MP), the sessions can be independent.

The summary of the comparison is listed in Table 2. We may also include other evaluation criteria in order to investigate the impact of adding links on the individual or overall performance. From the comparison result, the suitable models can be adopted and adapted based on the problem properties.

6. Conclusion and future directions

With the advent of the Internet technology, the information crawling/gathering on the Web is highly demanded and important in various applications. For example, users need to surf on the Web for sourcing in procurement process. However, dynamic traffic and poorly organized Web pages lead the users to navigate through irrelevant and repeated pages. In this paper, we adopt a stochastic searching approach to provide users with dynamic guidance for information access on the Web. We consider five models: inexperienced surfers on guidance-less sites, experienced surfers on guidance-less sites, sites with the mean-length guidance, sites with the simulation result, we conclude that the dynamic guidance indeed improve the access performance. We also propose a reformulation policy to re-sequence the hyperlinks as highlighted guidance in a page.

We can further extend this research to the following directions:

(1) *Multiple-destination*. The dynamic guidance algorithm can be extended to cover a minimum-spanning tree, if the user visits multiple pages.
(2) *Oscillation avoidance*. Since the dynamic guidance takes into account the traffic condition, we may need to add a "stopping rule" or a learning mechanism to avoid the oscillation in selecting the remaining path.
(3) *Enabled cache*. The traverse information can be kept in the cache for "go-back" function. In this case, we need to add a reverse link with zero length or change its length to zero if it exists in the original graph.
(4) *Dynamic information content*. If the pages are generated dynamically (such as ASP), we can split/merge the pages and re-organize information links/content.
(5) *Personalized information space*. The dynamic information content can be further personalized that each user surfs on the customized Web information space based on his preference requirement.

Acknowledgement

It is a pleasure to thank Ms. Jenny Zhong for implementing part of the system in the case study during her stay at HKUST. This research is partially sponsored by Taran Eastman Publishing Ltd. (TEIL 99/00.EG01) and by The University of Hong Kong (Research Initiation Grant 02-04). The authors would like to acknowledge the helpful comments by the anonymous reviewers.

References

[1] Perkowitz, M and Etzioni, O. (2000) Adaptive Web Sites. *Communication of ACM*, Vol. 43, No. 8, August.

[2] Perkowitz, M and Etzioni, O. (2000) Towards adaptive Web sites: conceptual framework and case study. *Artificial Intelligence*. Vol. 118, No.1-2, pp. 245-75

[3] Yan, T.W., Jacobsen, M., Garcia-Molina, H., and Dayal, U. (1996) From user access patterns to dynamic hypertext linking. *Computer Networks & ISDN Systems*, vol.28, no.7-11, pp.1007-14.

[4] Wang, Z., Siew, C.K., and Yi, X. (2000) A new personalized filtering model in Internet Commerce, *Proceedings of SSGRR (Scuola Superiore G. Reiss Romoli)*, Rome, Italy.

[5] Cheung, D.W., Kao, B. and Lee, J. (1998) Discovering user access patterns on the World Wide Web. *Knowledge-Based Systems*, vol.10, no.7, pp.463-70.

[6] Chen, P.M and Kuo, F.C. (2000) An Information Retrieval System based on User Profile. *The Journal of System and Software*, vol. 54, pp3-8.

[7] Chang, C.H., Hun, C.C. and Hou, C.L. (1998) Exploiting hyperlinks for automatic information discovery on the WWW. *Proceedings of 10th IEEE International Conference on Tools with AI*, pp.156-63.

[8] Yang, C.C., Yen, J. and Chen, H. (2000) Intelligent Internet Searching Agent Based on Hybrid Simulated Annealing. *Decision Support Systems*. Vol.28 pp.269-277

[9] Tu, H.C. and Hsiang, J. (2000) An Architecture and Category Knowledge for Intelligent Information Retrieval Agents. *Decision Support Systems*. Vol. 28, pp.255-268

[10] Teng, S.-H., Lu, Q., Eichstaedt, M., Ford, D. and Lehman, T. (1999), Collaborative team crawling: information gathering/processing over Internet, *Hawaii International Conference on System Sciences: HICSS32*.

[11] Joachims, T., Freitag, D. and Mitchell, T. (1997) WebWatcher: A Tour Guide for the World Wide Web, *Proceedings of IJCAI-97*, Nagoya, Janpan, pp770-775

[12] Liaberman, H. (1995) Letizia: an agent that assists Web browsing. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (IJCAI-95). vol. 1, pp.924-929

[13] Berghel, H., Berleant, D., Foy, T., and McGuire, M. (1999) Cyberbrowsing: information customization on the Web. *Journal of the American Society for Information Science*, vol.50, no.6, pp.505-13

[14] Lin, I.-Y., Huang, X.M. and Chen, M.S. (1999) Capturing user access patterns in the Web for data mining. *Proceedings 11th International Conference on Tools with Artificial Intelligence*, pp.345-8.

[15] Richardson, O. (2000) Gathering accurate client information from World Wide Web sites. *Interacting with Computers*. Vol.12, no.6, pp.615-22.

[16] Gibson, D., Kleinberg, J. and Raghavan, P. (1998) Structural Analysis of the World Wide Web. *WWW Consortium Web Characterization Workshop*, November.

[17] Chakrabarti, S., Dom, B., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Kleinberg, J.M. and Gibson, D. (1999) Hypersearching the Web, *Scientific American*, June.

[18] Chakrabarti, S., Dom, B., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D. and Kleinberg, J.M. (1999) Mining the Web's Link Structure. *IEEE Computer*, 32(8): 60-67

[19] Kleinberg, J., Kumar, S.R., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999) The Web as a graph: Measurements, models and methods. *International Conference on Combinatorics and Computing*.

[20] Sarukkai, R.R. (2000). Link Prediction and Path Analysis using Markov Chains. *Computer Network*, vol. 33 pp377-386.

[21] Levene, M., Borges, J. and Loizou, G. (2001) Zipf's law for web surfers. *Knowledge and Information Systems an International Journal*, 3, 120-129.

[22] Levene, M. and Loizou G. (1999) A probabilistic approach to navigation in Hypertext. *Information Sciences*, 114, 165-186.

[23] Levene, M. and Loizou G. (1999) Navigation in Hypertext is easy only sometimes. *SIAM Journal on Computing*, 29, 728-760.

[24] Kumar, R., Rajagopalan, S., Sivakumar, D., Tomkins A. and Upfal, E. (2000) Stochastic models for the web graph. *Proceedings of the IEEE Symposium on Foundations of Computer Science*.

[25] Levene, M., Fenner, T., Loizou, G. and Wheeldon, R. (2002) A stochastic model for the evolution of the web. *Condensed Matter Archive*, cond-mat/0110016 v2.

[26] Zin, N. and Levene, M. (1999) Constructing web views from automated navigation sessions. In *ACM Digital Library WOWS*, Berkeley, Ca., August, pp. 54-58.

[27] Glover, F., Klingman, D. and Phillips, N. (1985) A New Polynomially Bounded Shortest Path Algorithm, *Operations Research*, 33, 65-73.

[28] Shier, D. and Witzfall, C. (1981) Properties of Labeling Methods for Determining Shortest Path Trees, *J. Res. Natl. Bur. Stand*., 86, 317-330.

[29] Psaraftis, H.N. and Tsitsiklis, J.N. (1993) Dynamic Shortest Path in Acyclic Networks with Markovian Arc Costs, *Operations Research*, 41, 91-101.

[30] Geetha, S. and Nair, K.P.K. (1993) On Stochastic Spanning Tree Problem, *Networks*, 23 675-679.

[31] Polychronopoulos. G. H. and Tsitsiklis, J.N. (1996) Stochastic Shortest Path Problems with Recourse, *Networks*, 27, 133-143.

[32] Cheung, R. K. (1998) Iterative Methods for Dynamic Stochastic Shortest Path Problems, *Naval Research Logistics*, 45, 769-789.

[33] Wolff, R. W. (1989) *Stochastic Modeling and the Theory of Queues*, Prentice-Hall, New Jersey.

[34] Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993) *Network Flows - Theory, Algorithms, and Applications*, Prentice-Hall International, New Jersey.