Optimal Assignment of Bookmarks to Web Pages

Jurek Czyzowicz [*‖] Evangelos Kranakis [†‡‖] Danny Krizanc [¶‡‖] Andrzej Pelc [*‡‖]
Miguel Vargas Martin [†‡§]

Abstract

Consider a web domain consisting of a collection $V = \{v_1, ..., v_n\}$ of web pages connected by hyperlinks. Assume that there exists a directed path of hyperlinks from the home page r to any other page of the collection. Let p_i be the probability that a user wants to access page v_i. A *bookmark* is an additional hyperlink from the home page r to any other page. We want to add k bookmarks to this web page collection, so as to minimize the expected access time from r, measured by the length of the shortest path of hyperlinks. We show that this optimization problem is NP-hard in general, even for uniform probability distribution, and solve it in $O(k)$ time when the domain is a complete binary tree with uniform distribution and $k \leq \sqrt{n+1}$.

1 Introduction

The remarkable and unregulated growth of the World-Wide Web has produced a huge and complex network with unpredictable and often slow response times. This network can be modeled as a directed graph, whose nodes are documents and edges are hyperlinks (URLs) pointing from one document to another. Its topology determines the web's connectivity from which the efficiency in locating information is derived. Because of its enormous size and of its dynamically changing structure it is impossible to catalogue all nodes and links. This weakness is also reflected by web engines in current use that must limit their coverage and cataloguing to a fraction of the total in order to achieve real-time performance guarantees [6].

An important way of improving the response rate of large, distributed documents is via a careful and methodical design of bookmark assignments to related web pages. This is the approach we follow in the present paper. Consider a part of the web called a *web domain*, consisting of a collection $V = \{v_1, ..., v_n\}$ of web pages connected by hyperlinks. These links have been placed a priori by design in the initial construction of the web pages. Assume that there exists a directed path of

*Département d'Informatique, Université du Québec à Hull, Hull, Québec J8X 3X7, Canada. Email: {Czyzowicz,Andrzej_Pelc}@uqah.uquebec.ca.

†School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada. E-mail: {kranakis,mvargas}@scs.carleton.ca.

¶Department of Mathematics, Wesleyan University, Middletown CT 06459, USA. Email: dkrizanc@caucus.cs.wesleyan.edu.

‖Research supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

‡Research supported in part by MITACS (Mathematics of Information Technology and Complex Systems) NCE (National Center of Excellence) grant.

§Research supported in part by CONACYT (Science and Technology Council of Mexico).

hyperlinks from the *home page* r to any other page of the collection. Let p_i be the probability that a user, currently located at r, wants to access page v_i. A *bookmark* is an additional hyperlink from the home page r to any other page of the domain. Assuming this we pose the following k-Bookmark Assignment Problem (k-BAP): Find an assignment of k bookmarks in a web domain with known access probabilities, that will minimize the expected number of steps to reach any page of the domain from the home page.

1.1 Related work and results of the paper

An equivalent problem concerns the placement of k web proxies in the internet. This latter problem has been studied in [8, 7] and the optimal placement of k proxies is equivalent to our problem k-BAP. In partcular, in [8] the authors provide a dynamic programming algorithm on an n-node line with running time $O(n^2k)$ and in [7] they provide a dynamic programming algorithm that solves the same problem on a tree of n nodes with running time $O(n^3k^2)$. A different problem is the hotlink assignment problem that was studied in [2]: by adding hotlinks, at most one per page, we are interested in minimizing the expected number of steps needed to visit from the home page a selected set of web pages.

In this paper we prove an NP-hardness result on general directed graphs and an optimal placement algorithm on complete binary trees. More specifically, our results concerning the k-BAP for networks with n nodes are the following.

1. k-BAP is NP-hard for arbitrary directed graphs (even with uniform distribution of access probabilities).

2. k-BAP can be solved in time $O(k)$ for complete binary trees with uniform distribution of access probabilities, if $k \leq \sqrt{n+1}$. Moreover, we give a complete characterization of optimal bookmark sets in this case.

Thus, our work extends the work of [7] not only by providing a more efficient solution when the distribution is uniform but also by giving the complete characterization of optimal bookmark assignments in this case.

1.2 Terminology

Let $G = (V, E)$ be an arbitrary directed graph, representing a web domain, and r a distinguished node of G called the *root*, representing the home page. Assume that there exists a directed path from r to any other node of G. Let $V = \{1, ..., n\}$ and let $(p_1, ..., p_n)$ be an arbitrary probability distribution over V, i.e., $p_i \geq 0$ and $p_1 + \cdots + p_n = 1$. p_i is the access probability to page i. For any node v, let $d_G(v)$ denote the *distance* in G from r to v, i.e., the length l of the shortest directed path $(r = v_0, ..., v_l = v)$. The *cost* of the graph G is defined as $\mathcal{C}(G) = \sum_{v=1}^{n} d_G(v)p_v$. A *bookmark set* on the graph G is any subset B of V. (Thus we identify the hyperlink from r to v with the node v itself.) The *augmentation* of graph G by the bookmark set B is the graph $G_B = (V, E')$, where $E' = E \cup \{(r, b) : b \in B\}$. The *gain* $\mathcal{G}(B)$ of the bookmark set B is the difference $\mathcal{C}(G) - \mathcal{C}(G_B)$, where G_B is the augmentation of G by B. A bookmark set of size k on a graph G is *optimal*, if $\mathcal{G}(B) \geq \mathcal{G}(B')$, for any bookmark set B' of size k on G.

In the special case of the uniform probability distribution $p_1 = ... = p_n = 1/n$ we define the *charge* of the graph G as $c(G) = \sum_{v=1}^{n} d_G(v)$ and the *value* $\mathcal{V}(B)$ of a bookmark set B as the difference $c(G) - c(G_B)$, where G_B is the augmentation of G by B. Hence, for uniform distributions, the charge is n times the cost and the value is n times the gain. We will also define the value of B for node v, $\mathcal{V}_v(B) = d_G(v) - d_{G_B}(v)$. Obviously, $\mathcal{V}(B) = \sum_{v=1}^{n} \mathcal{V}_v(B)$.

2 Arbitrary graphs

In this section we prove that the problem of optimal assignment of a given number of bookmarks in arbitrary directed graphs is NP-hard, even for uniform distributions. More precisely, consider the following optimization problem.

BOOKMARK ASSIGNMENT

Instance: Directed graph $G = (V, E)$ and positive integer $k \leq |V|$.

Task: Find a bookmark set $B \subseteq V$ of size k maximizing $\mathcal{V}(B)$.

Theorem 1 *The problem BOOKMARK ASSIGNMENT is NP-hard.*

Proof: The transformation is from the following NP-complete decision problem [4]:

MINIMUM COVER

Instance: Collection C of subsets of a finite set S, positive integer $K \leq |C|$.

Question: Does C contain a cover for S of size K or less, i.e., a subset $C' \subseteq C$ with $|C'| \leq K$, such that every element of S belongs to at least one member of C'? Consider an instance of MINIMUM COVER, let $C = \{c_1, ..., c_t\}$ and construct the following directed graph $G = (V, E)$. Let $A = \{a_1, ..., a_t\}$, $X = \{x_1, ..., x_t\}$ and $Y = \{y_1, ..., y_t\}$ be sets such that A, X, Y, S are pairwise disjoint. Let r be an element outside of $A \cup X \cup Y \cup S$. Define $V = A \cup X \cup Y \cup S \cup \{r\}$. The set E of arcs is defined as follows (cf. Fig 1). $E_1 = \{(a_i, s) : s \in c_i, i = 1, ..., t\}$, $E_2 = \{(x_i, a_i) : i = 1, ..., t\}$, $E_3 = \{(y_i, x_i) : i = 1, ..., t\}$, $E_4 = \{(r, y_i) : i = 1, ..., t\}$. Finally, $E = E_1 \cup E_2 \cup E_3 \cup E_4$. The

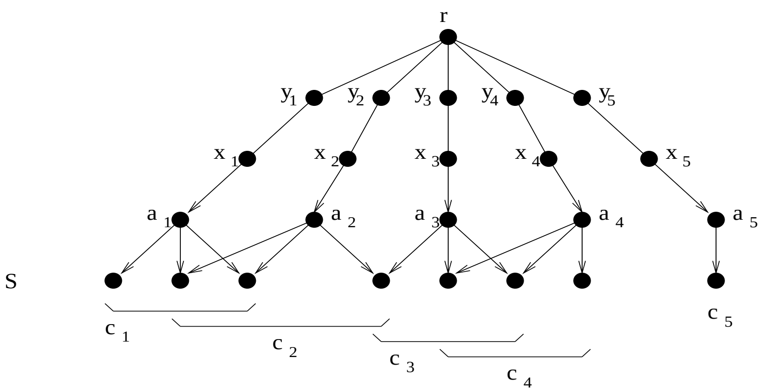

Figure 1: An instance of BOOKMARK ASSIGNMENT

construction of graph $G = (V, E)$ can be done in polynomial time. For the rest of the proof see the appendix. ∎

3 Bookmarks in Complete Binary Trees

As already mentioned in the intorduction, in [7] the authors provide a dynamic programming algorithm that solves k-BAP on an arbitrary tree of n nodes with running time $O(n^3k^2)$. In this section we show how to compute the optimal solution to k-BAP on a complete binary tree.

Consider a tree G rooted at r, oriented from r to the leaves. Suppose that G has been augmented by a bookmark set B. We say that a node $x \in V$ *dominates* node $v \in V$, if on the path $(x = v_0, v_1, \cdots, v_l = v)$ only x may belong to B. When a bookmark b dominates v, the shortest path from root r to v uses the edge (r, b). The set $\mathcal{D}(b)$ of all nodes of V dominated by b is called the *domain* of b. We say that two bookmarks b_1 and b_2 are *independent* if there is no oriented path from one of them to the other. Observe that domains of two bookmarks b_i and b_j are always disjoint and the family of domains of the set $B \cup r$ partitions V. We say that B *covers* G if each leaf v of G belongs to the domain of some $b_i \in B$. A bookmark $b \in B$ is called *exposed* in G_B if its domain does not contain any other bookmark of B. For $p \geq 0$, the *level* p of tree G consists of all nodes at distance p from r. If $p < q$ we say that p is *above* q or q is *deeper* than p. The *depth* of the tree is the number of its deepest level.

We construct an optimal assignment of k bookmarks in the complete binary tree of $n \geq 15$ nodes with uniform probability distribution, under some condition relating k to n. First observe that if $k \leq 4$ then the optimal assignment of bookmarks is to place all of them on level 2. (Putting a bookmark on level 0 or 1 would mean wasting it.) Hence we may assume $k > 4$.

Theorem 2 *For a complete binary tree G of $n \geq 15$ nodes, a set B of k bookmarks, such that $4 < k \leq \sqrt{n+1}$, is optimal, if and only if all of the following conditions hold:*

1. *B covers G,*
2. *all bookmarks in B are independent,*
3. *all bookmarks in B are placed on two consecutive levels of G.*

The existence of a set of bookmarks satisfying the above conditions is guaranteed by the following lemma:

Lemma 3 *For a complete binary tree G on n nodes and for any number $k \leq n$, there exists a set of k independent bookmarks covering G, placed on at most two consecutive levels of G.*

Proof: Let $p = \lfloor \log(k) \rfloor$. If we take any set S of $2^{p+1} - k$ bookmarks on level p, we can place exactly $2k - 2^{p+1}$ bookmarks independent on S on level $p + 1$. They all cover G. ∎

Note that the distribution of bookmarks given in Lemma 3 is unique up to the choice of its subset on level p. Thus in the remainder of this section we will suppose without loss of generality that in the set of bookmarks from Theorem 2 the bookmarks from the higher level are the leftmost nodes of G on this level (cf. Fig. 2). Theorem 2 implies the following corollary.

Corollary 4 *Under assumptions of Theorem 2 an optimal set of bookmarks can be constructed in time $O(k)$.*

In order to prove Theorem 2 we formulate several lemmas showing which transformations of a bookmark set increase its value.

Lemma 5 *Let B_1 be a set of bookmarks defined for a tree G. Suppose that B_1 has been altered by replacing some of its bookmarks $b_{i_1}, b_{i_2}, \ldots, b_{i_m}$ by $b_{j_1}, b_{j_2}, \ldots, b_{j_m}$ thus forming a new bookmark set B_2. For each node v, which is not in the domain of any node from $X = \{b_{i_1}, b_{i_1}, \ldots, b_{i_1}, b_{j_1}, b_{j_1}, \ldots, b_{j_1}\}$ we have $\mathcal{V}_v(B_1) = \mathcal{V}_v(B_2)$.*

Proof: Obvious, since the shortest oriented path from r to v in G_{B_1} and G_{B_2} must use exactly the same edges. ∎

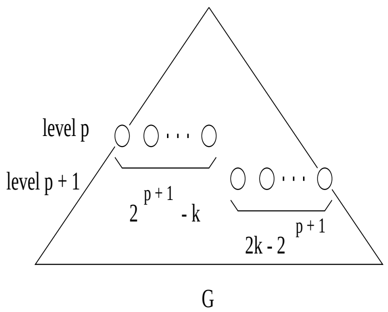

Figure 2: The set of bookmarks from Theorem 2.

Lemma 5 states that, when replacing one bookmark set by another, $\mathcal{V}_v(B)$ may change only for the nodes v which are in the subtrees rooted at the nodes of X. We will denote $\mathcal{V}_{-X}(B)$ the value of the bookmark set for the graph G diminished by the nodes of X and their descendants. The statement of Lemma 5 says that $\mathcal{V}_{-X}(B_1) = \mathcal{V}_{-X}(B_2)$.

Lemma 6 ***(lift).*** *Suppose that for a set of bookmarks B on G there exists a node x on level l, such that, among the descendants of x, the only bookmark $y \in B$ is a child of x. Then for $B' = B \setminus \{y\} \cup \{x\}$, we have $\mathcal{V}(B') > \mathcal{V}(B)$.*

Proof: Let w be the number of nodes in the subtree rooted at y and s be the distance from r to x in G_B, see Fig. 3. Note that $s < l$ if there is a bookmark on the oriented path from r to x,

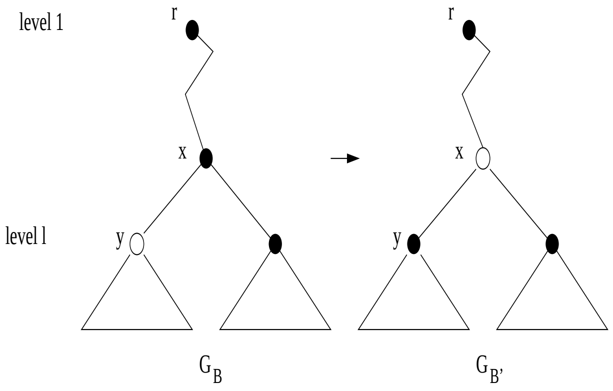

Figure 3: For Lemma lift.

otherwise $s = l$. By Lemma 5, $\mathcal{V}_{-\{x\}}(B) = \mathcal{V}_{-\{x\}}(B')$ because $\mathcal{V}_v(B) = \mathcal{V}_v(B')$, for each node v not in the subtree rooted at x. Therefore, to evaluate the change in the value function between the sets B and B' we should consider only nodes in the subtree rooted at x. Hence, $\mathcal{V}_v(B) = l$, for all w nodes in the subtree rooted at y and $\mathcal{V}_v(B) = l - s$, for x and all remaining w descendants of x, while $\mathcal{V}_v(B') = l - 1$, for all $2w + 1$ nodes in the subtree rooted at x. Since $l \geq s \geq 2$ we conclude that $\mathcal{V}(B) = \mathcal{V}_{-\{x\}}(B) + lw + (l - s)(w + 1) < \mathcal{V}_{-\{x\}}(B) + (l - 1)(2w + 1) = \mathcal{V}(B')$. ∎

Lemma 7 ***(adopt).*** *Suppose that G_B contains two nodes x_1 and x_2 at the same level l, such that $x_1 \in B$ and no other node in the subtrees rooted at x_1 and x_2 belongs to B. Suppose that the last* *bookmark on the oriented path from r to x_1 in the tree G is the node $b_1 \in B$ on level l_1 in G. Suppose as well, that if bookmark $b_2 \in B$ on level l_2 dominates x_2, we have $l_1 > l_2$. Then for $B' = B \setminus \{x_1\} \cup \{x_2\}$ we have $\mathcal{V}(B') > \mathcal{V}(B)$ (cf. Fig. 4).*

Proof: Let w be the number of nodes in the subtrees rooted at x_1 or x_2. Let $\bar{b_2}$ denote the bookmark

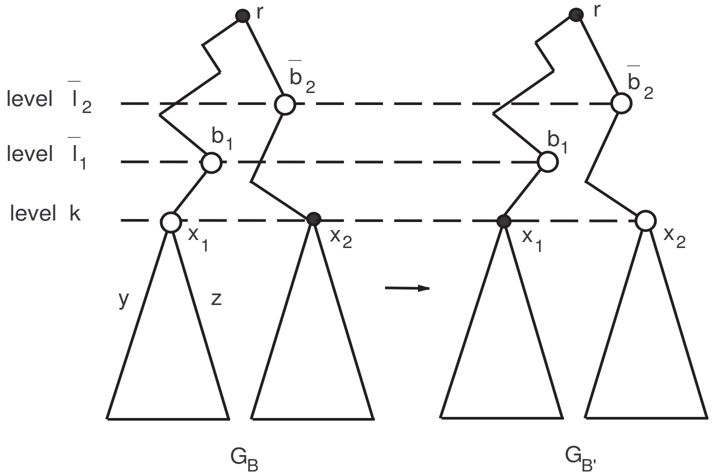

Figure 4: For Lemma adopt.

dominating x_2 or $\bar{b_2} = r$ if such a bookmark does not exist, and let $\bar{l_2}$ be the level of $\bar{b_2}$ in G. Again, $\mathcal{V}_{-\{x_1,x_2\}}(B) = \mathcal{V}_{-\{x_1,x_2\}}(B')$. We have $\mathcal{V}_v(B) = l-1$, for each node v, descendant of x_1 and $\mathcal{V}_v(B) = \bar{l_2} - 1$, for each v, descendant of x_2. Similarly, $\mathcal{V}_v(B') = l_1 - 1$, for each v, descendant of x_1 and $\mathcal{V}_v(B') = l-1$, for each v, descendant of x_2. Hence $\mathcal{V}(B) = \mathcal{V}_{-\{x_1,x_2\}}(B) + w(l-1) + w(\bar{l_2}-1) < \mathcal{V}_{-\{x_1,x_2\}}(B') + w(l_1-1) + w(l-1) = \mathcal{V}(B')$. ∎

Lemma 8 ***(spread).*** *Suppose that for a set of bookmarks B on G there exists a node x_1 on level $l \geq 2$ in G, such that the only bookmarks contained in the subtree of G rooted at x_1 are y and z - both children of x_1. Suppose as well that there exists another node x_2 on level l, such that the subtree rooted at x_2 contains no bookmarks. Then, for $B' = B \setminus \{y,z\} \cup \{x_1,x_2\}$ we have $\mathcal{V}(B') > \mathcal{V}(B)$.*

Proof: Let w be the number of nodes in the subtrees rooted at y or z and s_i - the distance from r to x_i in G_B, for $i = 1,2$, cf. Fig. 5. Again, $\mathcal{V}_{-\{x_1,x_2\}}(B) = \mathcal{V}_{-\{x_1,x_2\}}(B')$. For each node v, descendant of x_1 we have $\mathcal{V}_v(B) = l$, while $\mathcal{V}_{x_1}(B) = l - s_1$. For all w nodes in the subtree rooted at x_2 $\mathcal{V}_v(B) = l - s_2$. Similarly, $\mathcal{V}_v(B') = l-1$, for all $4w+2$ nodes in the subtrees rooted at x_1 and x_2. Since $s_i \geq 2$ we conclude that $\mathcal{V}(B) = \mathcal{V}_{-\{x_1,x_2\}}(B) + 2lw + (l-s_1) + (l-s_2)(2w+1) < \mathcal{V}_{-\{x_1,x_2\}}(B') + (l-1)(4w+2) = \mathcal{V}(B')$. ∎

Lemma 9 ***(floor).*** *If a set B of k bookmarks is optimal then each bookmark is placed on level $l = \lceil \log k \rceil$ of G or on a level above l.*

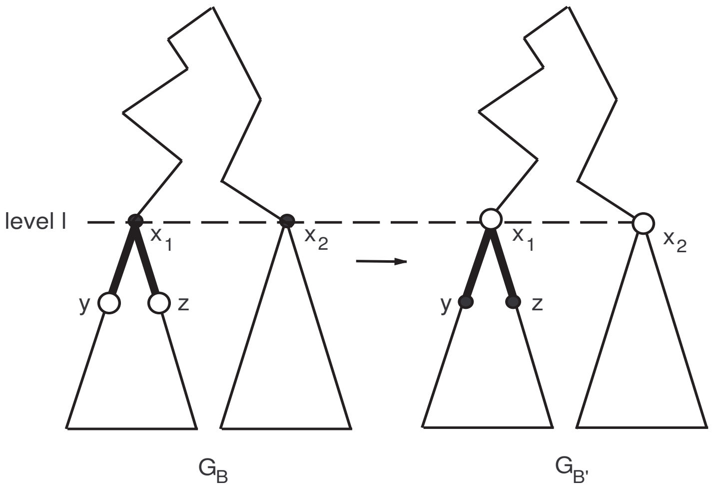

Figure 5: For Lemma spread.

Proof: Let b be a bookmark at the deepest level l_b among all the bookmarks of B. We will prove by induction, that if $l_b > \lceil \log k \rceil$, then we can always replace b by some $\bar{b}$, such that $\mathcal{V}(B) < \mathcal{V}(B \setminus \{b\} \cup \{\bar{b}\})$. There are three possible cases:

1. Neither the parent nor the sibling of b belongs to B. We can apply Lemma *lift* replacing b by $\bar{b}$ on a level above l in G.
2. The node a which is the parent of b belongs to B. As for the level l_a of a we have $l_a \geq \lceil \log k \rceil$. Hence there are at least $p = 2^{\lceil \log k \rceil}$ nodes on level l_a. As $p \geq k$, there exists a node u on level l_a, such that the subtree of G rooted at u contains no bookmarks. Thus the conditions of Lemma *adopt* are met, with $b_1 = a$, $x_1 = b$ and $x_2 = u$. We can choose $\bar{b} = u$.
3. The parent of b does not belong to B, but its sibling c does. Again, there must exist a node u on level l_x such that the subtree of G rooted at u contains no bookmarks and the conditions of Lemma *spread* are met with $y = b$, $z = c$ and $x_2 = u$. We take $\bar{b} = u$.

As in each case replacing b by $\bar{b}$ improves the value function of the boomark set, this concludes the proof. ∎

Lemma 10 ***(inherit).*** *Suppose that for a set B of k bookmarks, defined on the tree G of $n \geq 15$ nodes, there exists a pair of nodes y at level l and x the parent of y, such that $2 \leq l \leq \lceil \log k \rceil$ of G. Suppose as well, that y is the only bookmark among the descendants of x. If $k \leq \sqrt{n+1}$, then for $B' = B \setminus \{x\} \cup \{z\}$, where z is the sibling of y, we have $\mathcal{V}(B') > \mathcal{V}(B)$.*

Proof: Let w be the number of nodes in the subtree rooted at y and s - the distance from r to x in G_B, cf. Fig. 6. Again, $\mathcal{V}_{-\{x\}}(B) = \mathcal{V}_{-\{x\}}(B')$. For each node v, descendant of y we have $\mathcal{V}_v(B) = l-1$, while for all $w+1$ remaining nodes in the subtree rooted at x we have $\mathcal{V}_v(B) = l-2$. Similarly, $\mathcal{V}_v(B') = l-1$ for all $2w$ descendants of x, and $\mathcal{V}_x(B') = l-1-s$. We have to prove that

$$\mathcal{V}(B) = \mathcal{V}_{-\{x\}}(B) + (l-2)(w+1) + (l-1)w < \mathcal{V}_{-\{x\}}(B') + 2(l-1)w + (l-1-s) = \mathcal{V}(B')$$

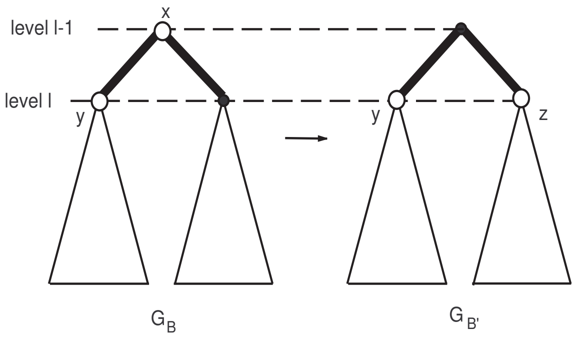

Figure 6: For Lemma inherit.

which is equivalent to

$$w + 1 > s \tag{1}$$

Note that $s \leq l$. Observe also that, since the depth of G equals $\log(n+1) - 1$, we have $w = 2^{\log(n+1)-l} - 1$. Therefore, since $k \leq \sqrt{n+1}$, which for $n \geq 15$ implies $\log k \leq \frac{\sqrt{n+1}}{2}$, we have $w + 1 \geq 2^{\log(n+1)-l} \geq 2^{\log(n+1)-\lceil \log k \rceil} \geq 2^{\log(n+1)-\log k - 1} = \frac{n+1}{2k} > \frac{\sqrt{n+1}}{4} \geq \frac{\log k}{2} > \lceil \log k \rceil \geq l \geq s$ ∎

Lemma 11 ***(expose).*** *Suppose that for a set B of k bookmarks on G there exists $b \in B$ at level $t \geq 3$, such that the only bookmarks contained in the subtree of G rooted at b, are $\tilde{B} = \{b_1, \cdots, b_m\}$, and*

1. *$\{b_1, \cdots, b_p\}$, where $1 \leq p \leq m$ are placed on level l of G,*
2. *$\{b_{p+1}, \cdots, b_m\}$ are placed on level $l+1$,*
3. *all bookmarks of $\tilde{B}$ are independent,*
4. *each leaf of G_B is in the domain of some bookmark from level l or $l+1$.*

If $k \leq \sqrt{n+1}$, then for $B' = B \setminus \{b, b_i\} \cup \{c_i, d_i\}$, where $b_i \in \tilde{B}$ is on level l and c_i, d_i are children of b_i we have $\mathcal{V}(B') > \mathcal{V}(B)$. See Fig. 7.

Proof: As $\mathcal{V}_{-\{b\}}(B) = \mathcal{V}_{-\{b\}}(B')$, we need only to compute the value function for the nodes in the subtree rooted at b. Let d be the depth of G and s - the length of the shortest oriented path in $G_{B'}$ from r to b. Observe that, there are $w = 2^{d-l} - 1$ nodes in the domain of each node $b_i, i = p+1, \ldots, m$, and for each of them $\mathcal{V}_v(B) = l$. Similarly $\mathcal{V}_v(B) = l - 1$, for each of $2w+1$ descendants of every node $b_i, i = 1, \ldots, p$. Finally, for v being one of $2^{l-t+1} - p - 1$ nodes in the domain of b we have $\mathcal{V}_v(B) = t - 1$. In the graph $G_{B'}$, there are $p - 1$ bookmarks which are descendants of b on level l and $m - p + 2$ bookmarks on level $l+1$. Similarly as in G_B, for each of $2w+1$ nodes in the domain of some bookmark on level l, we have $\mathcal{V}_v(B) = l - 1$ and for each of w nodes from the domain of a bookmark of level $l+1$, $\mathcal{V}_v(B) = l$. The subtree of $G_{B'}$ rooted at b has $2^{l-t+1} - p - 2$ nodes, which are not in the domain of any of $b_i, i = 1, \ldots, m$. For every such node $\mathcal{V}_v(B) = t - s$. Thus we have to prove that

$$\mathcal{V}(B) \quad = \quad \mathcal{V}_{-\{b\}}(B) + (t-1)(2^{l-t+1} - p - 1) + p(2w+1)(l-1) + (m-p)wl$$

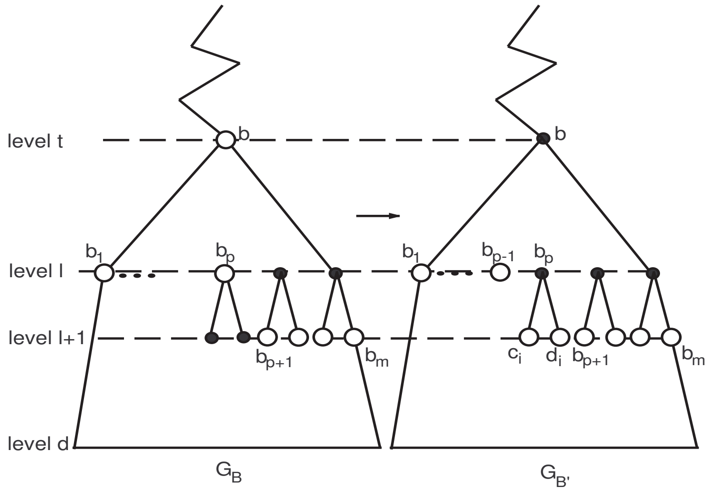

Figure 7: Exposing a node from a higher level.

$$
\begin{aligned}
&< \mathcal{V}_{-\{b\}}(B') + (t-s)(2^{l-t+1} - p) + \\
&\quad (p-1)(2w+1)(l-1) + (m-p+2)wl \\
&= \mathcal{V}(B')
\end{aligned}
$$

or, equivalently, that $(t-1)(2^{l-t+1} - p - 1) + (2w+1)(l-1) < (t-s)(2^{l-t+1} - p) + 2wl$. However, as $2 \le s \le t$ and $1 \le p \le m$, it is sufficient to prove a stronger condition substituting $s = t$ and $p = 1$. For these values, the above inequality becomes $(t-1)(2^{l-t+1} - 2) + (2w+1)(l-1) < 2wl$, which is equivalent to $(t-1)(2^{l-t+1} - 2) + l < 2w+1$. As $t \ge 2$, it is sufficient to prove the stronger condition $(t-1)2^{l-t+1} + (l-2) < 2w+1$. The function $f(t) = (t-1)2^{l-t+1} + l - 2$ attains its maximum when $f'(t) = 2^{l-t+1} - (t-1)2^{l-t+1} \ln 2 = 0$, i.e. when $t = \frac{1}{\ln 2} + 1$. Since t is a level number and f is an integer argument unimodal function, it is easy to verify that $f(t)$ obtains the same maximal value for $t = 2$ and $t = 3$. Hence, we can strengthen again the inequality we have to prove, getting $2^{l-1} + (l-1) < 2w+1$. Since $w = 2^{d-l} - 1$ and $n = 2^{d+1} - 1$, the condition becomes

$$2^{l-1} + (l-1) < (n+1)2^{-l} \tag{2}$$

However, since $2^{l-1} \ge l-1$, once again we can strengthen the condition obtaining $2^l < (n+1)2^{-l}$ or

$$2^{2l} < (n+1) \tag{3}$$

Observe that, since each leaf of G_B is in the domain of some bookmark from level l or $l+1$, each of 2^l nodes on level l or one of its children must belong to B. As $b \in B$, we identified at least $2^l + 1$ bookmarks belonging to B, proving that $k > 2^l$. Since $k \le \sqrt{n+1}$, we conclude $2^{2l} < k^2 \le (n+1)$ thus proving (3). ∎

3.1 Proof of Theorem 2

Let B be an optimal set of bookmarks. By Lemma *floor*, we can suppose that each bookmark $b \in B$ is on or above the *floor* level $l = \lceil \log k \rceil$. Take a bookmark $b \in B$ on the deepest level l_b. Let c be a sibling of b and let a be its parent. Observe that at most one node among a and c can belong to B, otherwise we could improve B by means of Lemma *expose*. However, in such a case, $a \notin B$, since we could apply Lemma *inherit*, again improving the value of B. If $c \notin B$, we could apply Lemma *lift*, once again getting a better bookmark set. So $c \in B$ and $a \notin B$. Similarly, for any other bookmark on level l_b its sibling must belong to B and its parent must not. Take any node x on level $l_b - 1$, whose children do not belong to B. $x \in B$, otherwise we could apply Lemma *spread* using $x_1 = a$ and $x_2 = x$. So for each node x on level l_b, either x itself or both its children belong to B. Consider the deepest level bookmark $y \in B$ above level $l_b - 1$. If y exists, y along with all the bookmarks of B which are descendants of y verify the conditions of Lemma *expose*. If y does not exist, the bookmarks of B are all independent, cover G and are placed on two consecutive levels $l_b - 1$ and l_b. ▮

The following proposition shows that the property of optimal bookmark sets described in Theorem 2 does not hold for larger numbers of bookmarks.

Proposition 12 *Let G be a complete binary tree with n nodes and let $k = 2\sqrt{n+1} + 1$. There exists a set B of k bookmarks in G such that:*

1. *B covers G,*
2. *all bookmarks in B are independent,*
3. *all bookmarks in B are placed on two consecutive levels of G,*

but B is not optimal.

References

[1] Abello, J., P. M. Pardalos and M. G. C. Resende, "On maximum clique problems in very large graphs". In External Memory Algorithms (J. Abello and J. Vitter, eds.), AMS-DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 50, 1999.

[2] P. Bose, J. Czyzowicz, L. Gąsieniec, E. Kranakis, D. Krizanc, A. Pelc, M. Vargas Martin, Strategies for Hotlink Assignments, in Proceedings of ISAAC 2000, held in Taipei, Taiwan, Dec. 2000, D. T. Lee and Shang-Hua Teng, eds., SVLNCS, Vol. 1969, pp. 23-34.

[3] M. Faloutsos, P. Faloutsos, C. Faloutsos, "On Power-Law Relationships of the Internet Topology", In proceedings of SIGCOM: IEEE Conference on Communication, 1999.

[4] M.R. Garey and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman and Co., New York, 1979.

[5] B. Hayes, "Graph Theory in Practice: Part I", American Scientist, January-February, Vol. 88, No 1, 2000.

[6] S. Lawrence, C. L. Giles, "Accessibility of information on the Web". Nature 400 (8 July 1999):107-109.

[7] B. Li, J. Golin, G. Italiano, X. Dong, and K. Sohraby, "On the Optimal Placement of Web Proxies in the Internet", in proceedings of INFOCOMM'99.

[8] B. Li, X. Dong, J. Golin, and K. Sohraby, "On the Optimal Placement of Web Proxies in the Internet: Linear Topology", in proceedings of 8th IFIP Conference on High Performance Networking, HPN'98.

Appendix

Proof of Theorem 1.
Now let $k = K \leq t$ and solve BOOKMARK ASSIGNMENT for the instance G, k. Let B be the obtained optimal bookmark set of size k on G. Hence B has maximum value among all sets of size k. Clearly $B \subseteq A \cup X \cup S$ (bookmarks placed in $\{r\} \cup Y$ would be wasted). We now construct a bookmark set $B^* \subseteq A \cup S$ whose value is not smaller than that of B. Suppose that $b = x_i \in B$. There are two cases.

Case 1. $a_i \in B$. Since $|B| \leq t$, there exists an index $j \leq t$ for which none of the elements a_j, x_j, y_j belongs to B. Replacing b by a_j increases the value of the bookmark set because a bookmark at a_j contributes at least 2 to the value, while a bookmark at b contributes only 1 in this case.

Case 2. $a_i \notin B$. Replacing b by a_i does not decrease the value of the bookmark set. Indeed, let α be the number of elements in c_i which do not belong to any set c_l with $a_l \in B$. Then a bookmark at b contributes at most $2 + \alpha$ to the value, while a bookmark at a_i contributes $2(1 + \alpha)$. Clearly, $2 + \alpha \leq 2(1 + \alpha)$.

The bookmark set B^* can be obtained from B in polynomial time by a repeated application of changes described above. It follows that if BOOKMARK ASSIGNMENT can be solved in polynomial time then a bookmark set $B^* \subseteq A \cup S$ of maximum value among all bookmark sets of size k can also be obtained in polynomial time.

Now observe that, in fact, $B^* \cap S = \emptyset$. Indeed, suppose that some $s \in S$ belongs to B^*. Since $|B^*| \leq t$, there exists an index $j \leq t$ for which none of the elements a_j, x_j, y_j belongs to B^*. There are two cases. If s belongs to some set c_l with $a_l \in B^*$ then replacing s by a_j increases the value of the bookmark set because a bookmark at s contributes only 1 to the value, while a bookmark at a_j contributes at least 2. If s does not belong to any set c_l with $a_l \in B^*$ then choose any c_l such that $s \in c_l$. In this case replacing s by a_l increases the value of the bookmark set because a bookmark at s contributes at most 3 to the value, while a bookmark at a_l contributes at least $2 \cdot 2 = 4$. Hence in both cases the value of B^* could be strictly increased which contradicts the optimality of B^*. This contradiction proves that $B^* \cap S = \emptyset$ and consequently $B^* \subseteq A$.

Suppose that $B^* = \{a_{i_1}, ..., a_{i_k}\}$. We have $\mathcal{V}(B^*) = 2(k + |c_{i_1} \cup ... \cup c_{i_k}|)$. It follows that the family $\{c_{i_1}, ..., c_{i_k}\}$ maximizes union size among all subfamilies of C of size k. Consequently, C has a subfamily C' of size k or less covering S, if and only if, $c_{i_1} \cup ... \cup c_{i_k} = S$. Clearly $c_{i_1} \cup ... \cup c_{i_k}$ can be computed in polynomial time knowing B^*. Hence MINIMUM COVER can be solved in polynomial time. This proves that BOOKMARK ASSIGNMENT is NP-hard.

Proof of Proposition 12.
Take a complete binary tree G of depth $d = 2l - 3$ and let B be a set of $k = 2^l + 1$ bookmarks. Note, that $k = 2\sqrt{n+1} + 1$. According to Lemma 3 we can place 2^l bookmarks on level l and one bookmark on level $(l+1)$, so that they are independent and cover G. To show that B is not optimal, consider B' - a set of $2^l + 1$ bookmarks consisting of all nodes of level l and of one bookmark on level three, cf. Fig. 8. We have

$$\begin{aligned} \mathcal{V}(B') - \mathcal{V}(B) &= [(l-1)(2^l)(2^{d-l+1} - 1) + 1(2^{l-2} - 1)] \\ &\quad -[(l-1)(2^l - 1)(2^{d-l+1} - 1) + 2l(2^{d-l} - 1)] \\ &= l + 1 > 0 \end{aligned}$$

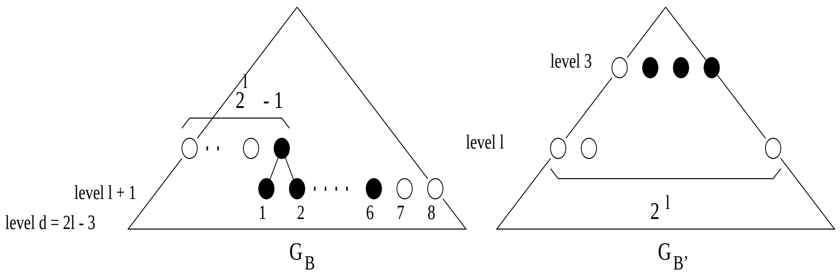

Figure 8: For Proposition 12.

thus proving that B is not optimal.