Two-Page Book Embedding of Trees under Vertex-Neighborhood Constraints

Shlomo Moran and Yaron Wolfstahl

Faculty of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel 32000

Abstract

We study the VLSI-related problem of embedding graphs in books. A book embedding of a graph \( G=(V,E) \) consists of two parts, namely, (1) an ordering of \( V \) along the spine of the book, and (2) an assignment of each \( e \in E \) to a page of the book, so that edges assigned to the same page do not intersect. In devising an embedding, one seeks to minimize the number of pages used.

A black/white (b/w) graph is a pair \( (G,U) \), where \( G \) is a graph and \( U \subseteq V \) is a subset of distinguished black vertices (the vertices of \( V - U \) are called white). A black/white (b/w) book embedding of a b/w graph \( (G,U) \) is a book embedding of \( G \), where the vertices of \( U \) are placed consecutively on the spine. The need for b/w embeddings may arise, for example, when the input ports of a multilayer VLSI chip are to be separated from the output ports.

In this paper we prove that every b/w tree admits a two-page b/w embedding. The proof takes the form of a linear time algorithm, which uses an extension of the unfolding technique introduced in [MW]. Combining this algorithm with the one in [MW] results in a linear time algorithm for optimal b/w embedding of trees.
1. Introduction

The embedding of a graph $G=(V,E)$ in a book consists of two parts. The first part is an ordering of $V$ along the spine of the book. The second part is an assignment of each edge $e \in E$ to a page of the book, such that if two edges are assigned to the same page, then they do not intersect. Given a graph $G$, the book embedding problem is that of finding a book embedding of $G$ with the minimum possible number of pages.

The notion of book embedding is useful in modeling layout problems arising in VLSI design (cf. [CLR]). Many of the basic properties of book embeddings were established in [BK]. In subsequent papers, several authors have investigated various aspects of book embeddings, establishing lower and upper bounds on the minimum number of pages required to embed some classes of graphs (cf. [CLR,H,R,Y]). We note that it is NP-complete to decide whether a given graph $G$ can be embedded in a 2-page book [W,CLR]. Also, given a graph $G=(V,E)$, an ordering of $V$, and an integer $k$, it is NP-complete to decide whether there is a $k$-page book embedding of $G$ with the vertices placed on the spine in the specified order [GJMP]. On the other hand, a graph admits a one-page book embedding iff it is outerplanar [BK], and this property is decidable in linear time [M].

A generalization of the book embedding problem, called the black/white (b/w) book embedding problem, was addressed in [MW]. In the generalized problem, a vertex-neighborhood constraint is imposed on the ordering of the vertices. Define a black/white (b/w) graph to be a pair $(G,U)$, where $G$ is a graph and $U \subseteq V$ is a subset of distinguished black vertices (the vertices of $V-U$ are called white). A black/white (b/w) book embedding of a b/w graph $(G,U)$ is a book embedding of $G$, where the vertices of $U$ are placed consecutively. Given a b/w graph $(G,U)$, the b/w book embedding problem is that of finding a b/w book embedding of $(G,U)$, such that the number of pages is minimized. The need for b/w embeddings may arise, for example, when the input ports of a multilayer VLSI chip are to be separated from the output ports. Also, in certain applications external wires are used to connect some vertices in order to, say, meet some connectivity property (cf. [ET]). Putting those vertices consecutively serves to reduce the total wire length, an important consideration in VLSI design.

In [MW], the authors introduced a technique called unfolding for constructing good b/w embeddings. Using this technique, an algorithm that constructs a one-page b/w embedding of a b/w graph $(G,U)$, provided such embedding exists, is developed there. In this paper we prove that every b/w tree admits a two-page b/w embedding. The proof takes the form of a linear time algorithm, which uses an extension of the unfolding technique. Combining this algorithm with the one in [MW] results in a linear time algorithm for optimal b/w embedding of trees.

The paper is organized as follows. The formal framework is described in Section 2. In Sections 3 we present the algorithm for b/w embedding of any b/w-tree in a two-page book, and discuss its optimality A sum-
mary is given in Section 4.

2. The Formal Framework

The following definitions provide a formal framework for the rest of this paper.

A book consists of a spine and a number of pages. The spine of the book is a line. For simple exposition, view the spine as being horizontal. Each page of the book is a half-plane that has the spine as its boundary. Thus, any half-plane is a 1-page book, and a plane with a distinguished horizontal line is a 2-page book.

Let $G = (V, E)$ be a graph, and let $\rho$ be a linear ordering of $V$. For each $v \in V$, denote the rank of $v$ with respect to $\rho$ by $\rho(v)$. Let $(v_i, v_j)$ and $(v_k, v_l)$ be two edges of $G$ where $\rho(v_i) < \rho(v_j)$ and $\rho(v_k) < \rho(v_l)$. If either $\rho(v_i) < \rho(v_k) < \rho(v_j) < \rho(v_l)$ or $\rho(v_k) < \rho(v_i) < \rho(v_l) < \rho(v_j)$, we say that $(v_i, v_j)$ and $(v_k, v_l)$ intersect, or are incompatible (w.r.t. $\rho$). Otherwise, we say that $(v_i, v_j)$ and $(v_k, v_l)$ are compatible. A set $E' \subseteq E$ is compatible if any two edges $e_1, e_2 \in E'$ are compatible.

A $k$-page book embedding of $G = (V, E)$ may be formally defined as a pair $E = (\rho, \pi)$, where $\rho$ is an ordering of $V$ along the spine and $\pi$ is a partition of $E$ into $k$ compatible sets. Each page of the book is said to accommodate one of these compatible sets. An edge accommodated by some page is also said to be assigned to that page.

A $k$-page b/w book embedding of the b/w graph $(G, U)$ is a $k$-page book embedding of $G$, say $E = (\rho, \pi)$, where the black vertices are consecutive in $\rho$. A basic property of book embeddings is that the order of the vertices on the spine can be viewed as a circular order. That is, if $E = (\rho, \pi)$ is a $k$-page book embedding of $G$, and $\rho'$ is a circular permutation of $\rho$, then $E' = (\rho', \pi)$ is also a $k$-page book embedding of $G$.

We say that a set of vertices $S \subseteq V$ is equally-colored if the vertices of $S$ are all either black or white. Vertices that are not equally-colored are said to be oppositely-colored.

Let $G = (V, E)$ be a graph. A path in $G$ is either a single vertex or a sequence of distinct vertices $(v_1, v_2, \ldots, v_k)$ where $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq k-1$. Let $p_1 = (v_1, \ldots, v_k)$ and $p_2 = (w_1, \ldots, w_l)$ be two paths in $G$, where $(v_k, w_1) \in E$. Then $p_1 \cdot p_2$ is the path $(v_1, \ldots, v_k, w_1, \ldots, w_l)$. A path in a b/w graph is mixed if it contains oppositely-colored vertices. A block of a mixed path is a maximal set of consecutive equally-colored vertices.

In [MW], the authors introduced a technique called b/w unfolding for constructing good b/w embeddings. To demonstrate this technique, we next exemplify the unfolding of a b/w graph consisting of a single mixed path. Let $p = V_1, V_2, \ldots, V_n$ be the path underlying such a graph, where each $V_k = (v_1^k, \ldots, v_{n_k}^k)$ is a block. Define $\bar{V}_k$, $1 \leq k \leq n$, to be the path obtained by reversing $V_k$. In the sequel, the paths $\bar{V}_k, V_k$ are viewed as the linear orders of their respective vertices. Given two linear orders $\rho_1 = (v_1, \ldots, v_n)$ and $\rho_2 = (w_1, \ldots, w_n)$, define $\rho_1 \rho_2$ to
be the linear order \( \rho_1 \rho_2 = (v_1, \ldots, v_n, w_1, \ldots, w_n) \). A b/w unfolding of \( p = \vec{v}_1 \vec{v}_2 \cdots \vec{v}_n \) is the ordering \( \rho(p) = \vec{v}_2 \vec{v}_4 \vec{v}_1 \vec{v}_3 \vec{v}_2 \cdots \). Two immediate properties of \( \rho(p) \) are the following. First, equally-colored vertices of \( p \) are consecutive in \( \rho(p) \). Second, the edges on \( p \) are all compatible w.r.t. \( \rho(p) \). It follows that \( \rho(p) \) underlies a one-page b/w embedding of \( G \). Figure 1 depicts the unfolding.

3. b/w Embedding of b/w Trees in Two Pages

As mentioned in the introduction, deciding whether there is a two-page book embedding of a given graph is NP-complete. It immediately follows that the b/w book embedding problem is also NP-complete. On the other hand, one-page b/w embeddability of a given b/w graph is decidable in linear time [MW]. These results motivate the study of b/w embedding of graph classes that admit efficient one-page (non-b/w) book embeddings. A natural class to consider in this context is the class of trees [DT], which form a basic VLSI interconnection pattern (cf. [BL,U]). In this section, we show that every b/w tree admits a two-page b/w embedding.

Definition: Let \((T, U)\) be a b/w tree where \( T = (V, E) \), and let \( u \) be a vertex of \( T \). Consider \( T \) as being rooted at \( u \). The b/w-depth of a vertex \( v \in V \) is the number of blocks on the unique path connecting \( v \) and the root. The b/w-depth of \( T \) is the maximum b/w-depth of a vertex in \( T \). Let \( S \) be a maximal set of equally-colored vertices of b/w-depth \( k', k' > 1 \), that have a common ancestor of b/w-depth \( k-1 \). Observe that the lowest common ancestor of the vertices of \( S \), called \( r_S \), has b/w-depth \( k-1 \), and its color is opposite to that of \( S \). The subtree induced by \( S \cup \{r_S\} \) is a depth-\( k \) subtree. This subtree, rooted at \( r_S \), is denoted by \( T(r_S) = (V(r_S), E(r_S)) \). In addition, define the depth-1 subtree of \( T \) to be the subtree induced on the vertices of b/w-depth 1. Given a book embedding of \( T \), the spine distance between two vertices is the number of vertices placed between them on the spine.
Definition: Let $T=(V,E)$ be a tree, let $v_1 \in V$ be a vertex of $G$ and let $\rho$ be a linear order of $V-\{v_1\}$. We say that $\rho$ is $v_1$-canonical if $\rho'=(v_1)\cdot \rho$ underlies a one-page book embedding of $T$ (or, equivalently, if $\rho''=\rho'(v_1)$ underlies a one-page book embedding of $T$).

Given a tree $T=(V,E)$ and $v \in V$, a $v$-canonical ordering of $V-\{v\}$ always exist. This follows from the fact that $T$, being outerplanar, admits a one-page book embedding [BK], where the underlying vertex ordering can be circularly permuted so that $v$ is the leftmost/rightmost vertex. In fact, a $v$-canonical ordering of $T$ can be found in linear time using the algorithm of [M] for recognizing outerplanar graphs.

In theorem 1 below we show that every b/w tree admits a two-page b/w embedding. The proof takes the form of an unfolding-based algorithm that, given a b/w tree, produces the claimed embedding by means of unfolding a set of paths. An additional tool used by the algorithm is canonical ordering, defined above.

Theorem 1: Every b/w tree $(T,U)$ admits a two-page b/w embedding, achievable in time linear in the size of $T$.

Proof: An algorithm is presented which embeds every b/w tree in a two-page book. Throughout the algorithm, the spine of the book is viewed as an horizontal line. The tree will be viewed as a rooted tree, and the first vertex to be embedded will be the root. Black and white vertices will be placed by the algorithm to the left and right of the root, respectively. The two pages used by the algorithm will be referred to as the upper page and the lower page.

Informally, the algorithm places the vertices in an increasing order of their respective b/w-depth. The white vertices are placed to the right of the rightmost vertex currently on the spine. Similarly, the black vertices are placed to the left of the leftmost vertex currently on the spine. The vertices of each depth-$k$ subtree rooted at, say, $v$ are placed in a $v$-canonical ordering. Edges entering black vertices are assigned to the lower page, and edges entering white vertices are assigned to the upper page. The unfolding technique here is thus applied to the directed paths of $T$, in such a way that they are nested so that two pages suffice.

Following, is a formal description of the algorithm (see Figure 2 for an example).

Algorithm Two-page

Input: A b/w tree $(T,U)$, where $T=(V,E)$
Output: A two-page b/w embedding of $(T,U)$

Method:
1. Choose a white vertex $v \in V$ to be the root of $T$ and place it on the spine. Place the other vertices of the depth-1 subtree of $T$ to the right of $v$, in a $v$-canonical ordering. Set $k=1$, and let $D$ be the b/w-depth of $T$.
2. while $k<D$
3  do
4     Let $v_1^k, v_2^k, \ldots, v_n^k$ be the vertices of b/w-depth $k$ currently on the spine, in an increasing order of their spine distance from $r$.
5     For $i = 1$ to $n_k$
6         do /* $v_i^k, 1 \leq i \leq n_k$ are roots of depth-$(k+1)$ subtrees */
7             Let $T(v_i^k) = (V(v_i^k), E(v_i^k))$ the depth-$(k+1)$ subtree rooted at $v_i^k$.
8         If $k$ is odd, then place $V(v_i^k) - \{v_i^k\}$ to left of the leftmost vertex currently on the spine, in a $v_i^k$-canonical ordering, and assign $E(v_i^k)$ to the lower page. /* $v_i^k$ is white, $V(v_i^k) - \{v_i^k\}$ are black */
9             If $k$ is even, then place $V(v_i^k) - \{v_i^k\}$ to right of the rightmost vertex currently on the spine, in a $v_i^k$-canonical ordering, and assign $E(v_i^k)$ to the upper page. /* $v_i^k$ is black, $V(v_i^k) - \{v_i^k\}$ are white */
10        od
11     Set $k \leftarrow k + 1$.
12  od

Insert Figure 2 here

Analysis. Note that the algorithm places the white vertices to the right of the black ones, and uses only two pages. We next prove, by induction on the b/w-depth of the vertices placed by the algorithm, that the edges assigned to each page are compatible. Let $T_k$ be the subtree induced by the vertices of b/w-depth at most $k$.

The induction claim states that for each $k, 1 \leq k \leq D$, the algorithm embeds $T_k$ in a two-page book.

The induction basis. It is easily seen that $T_1$ satisfies the claim. In fact, the edges of $T_1$ are all assigned to the upper page (see edges $((1,5),(1,6),(6,9))$ in Figure 2-b).

The induction step. The set of vertices of b/w-depth $k$ $[k+1]$ is hereafter denoted by $D_k$ [$D_{k+1}$]. Suppose that the induction claim holds for $T_k$. We next show that the algorithm places the vertices $D_{k+1}$, and assigns the corresponding edges, so that the claim remains valid for $T_{k+1}$. Assume that $k$ is odd, so $D_k$ are white. The algorithm placed $D_k$ consecutively, to the right of the rightmost vertex of $T_{k-1}$, so $D_k$ are the rightmost vertices of $T_k$. Let $D_k = \{v_1^k, v_2^k, \ldots, v_n^k \}$, in an increasing order of their spine distance from $r$. Now, $D_{k+1}$ are exactly the black vertices of the depth-$(k+1)$ subtrees rooted at the $v_i^k$, namely the subtrees $T(v_i^k) = (E(v_i^k), V(v_i^k))$, $1 \leq i \leq n_k$. Before placing $D_{k+1}$ there are no edges on the lower page incident to any of the vertices $v_i^k$, since the edges incident to the $v_i^k$ are assigned to the upper page. Note that $D_{k+1}$ are all placed to the left of the leftmost vertex of $T_k$, and the edges connecting them with vertices of $D_k$ are all assigned to the lower page. It follows that every edge incident to a vertex of $D_{k+1}$ is compatible with the lower-page edges of $T_k$. The latter are compatible, by the induction hypothesis. To prove the induction claim for $T_{k+1}$, it thus suffices to show that the edges entering vertices of $D_{k+1}$ are compatible. This is established as follows:
For each $T(v_f^k)$, the vertices $V(v_f^k)\setminus \{v_f^k\}$ are placed in a $v_f^k$-canonical ordering. Thus, each $E(v_f^k)$ is compatible, by definition of canonical ordering.

For different $i,j$ where $1 \leq i < j \leq n$, the vertices $V(v_f^k)\setminus \{v_f^k\}$ are placed to the left of the vertices $V(v_t^k)\setminus \{v_t^k\}$. Since $v_f^k$ was placed to the right of $v_t^k$, it follows that $E(v_f^k)$ is nested within $E(v_t^k)$. Hence, the edges of the depth-$(k+1)$ subtrees are compatible.

We thus conclude that the induction claim holds for $T_{k+1}$, assuming odd $k$. The claim for even $k$ is proved symmetrically.

The $b/w$-depth of each vertex of $T=(V,E)$, hence the depth-$k$ subtrees, $1 \leq k \leq D$, can be found in $O(|E|)$ time BFS [E]. The canonical ordering of each subtree $T(v_f^k)=(E(v_f^k),V(v_f^k))$ can be found in time $O(|E(v_f^k)|)$, using the aforementioned algorithm of [M]. Thus, finding the required canonical orderings of all the subtrees is $O(|E|)$. It follows that the overall embedding process can be implemented in $O(|E|)=O(|V|)$ time.

We next discuss the optimality of our algorithm.

Definition: Let $(T,U)$ be a b/w tree. A vertex $v \in V$ is a $U$-fork if $(T',U)$ contains three vertex-disjoint mixed paths, each of which does not contain $v$ and has its first vertex adjacent to $v$ (recall that a path in a $(T',U)$ is mixed if it contains oppositely-colored vertices). Figure 3 depicts a $U$-fork.

**Lemma 1 [MW]:** If $(T,U)$ contains a $U$-fork as a b/w subtree, then $(T,U)$ does not admit a one-page b/w embedding. Moreover, there is a linear time algorithm for b/w embedding of every b/w tree $(T,U)$ in a one-page book, provided $(T,U)$ does not contain a $U$-fork.

**Corollary:** There is a linear time algorithm that finds an optimal b/w embedding of b/w trees.

**Proof:** The claimed algorithm is obtained by executing simultaneously the algorithm of Theorem 1 and the algorithm of [MW].

4. Summary
Using an extension of the unfolding technique introduced in [MW], we have presented a linear time algorithm for b/w embedding of every b/w tree in a two-page book. Combining this algorithm with the one in [MW], a linear time algorithm for optimal b/w embedding of trees is obtained. We are currently studying b/w embedding of outerplanar graphs, where the unfolding technique also seems to be useful.

The difference between the basic unfolding technique [MW] and its extension used here is as follows. The basic technique involves the unfolding of a carefully-chosen single path, such that the rest of the graph can be embedded within a small number of pages. In the extended technique the graph is viewed as a set of paths that are unfolded in parallel, ensuring that these paths nest properly within a small number of pages. In general, the choice of the specific path(s) to unfold depends on the graph in hand, and is a key to the efficient use of unfolding. The reader interested in this technique is referred to [MW]; the b/w embedding algorithm developed there demonstrates yet another choice of the path to unfold.

References


Figure 1. Unfolding $p$

(In all the figures, black vertices are drawn as concentric circles; thus here $p$ starts in a white block).

Figure 2. Sample execution of Algorithm Two-page

a. A b/w tree $(T, U)$, where $U=\{2,3,4,7,11\}$,...

b. ... and the obtained b/w embedding, vertex 1 being the root.
Fig. 3 A b/w tree with a $U$-fork, where $U = \{1, 9, 13\}$. The three vertex-disjoint mixed paths are $(6, 5, ..., 1), (10, ..., 13)$ and $(8, 9)$. Throughout the paper, black vertices are drawn as two concentric circles.