UPWARD PARTITIONED BOOK EMBEDDING

Hugo A. Akitaya, Erik D. Demaine, Adam Hesterberg, and Quanquan C. Liu
Book Embedding

- Introduced in 1979 by Bernhart and Keinen [BK79]

- **Book Embedding**: Given undirected graph $G = (V, E)$:
  - Linear ordering, $\pi$, of vertices into spine
  - Disjoint partition of edges into sets so that each set can be embedded into page

- Pages join together to form book and **book thickness** is the minimum number of pages in any book embedding of $G$

- Applications in VLSI design, parallel process scheduling, others

Image from Wikipedia
Directed Graphs and Partitioned Problem

- Upward book embedding [HP97, HPT99]: Given directed DAG, embed the DAG such that the ordering on the spine is in topological order
Directed Graphs and Partitioned Problem

- **Upward book embedding** [HP97, HPT99]: Given directed DAG, embed the DAG such that the ordering on the spine is in topological order.

- **Upward partitioned book embedding (Upward)**: Given partition of edges into $k$ pages ($k$ disjoint sets of edges), linearly order vertices on spine.
Directed Graphs and Partitioned Problem

- **Upward book embedding** [HP97, HPT99]: Given directed DAG, embed the DAG such that the ordering on the spine is in topological order.

- **Upward partitioned book embedding (Upward)**: Given partition of edges into $k$ pages ($k$ disjoint sets of edges), linearly order vertices on spine.

- **Upward matching-partitioned book embedding (Matching)**: Edge partitions form a matching.
### Results: Old and New (All Partitioned)

<table>
<thead>
<tr>
<th>Type</th>
<th>(k = 1)</th>
<th>(k = 2)</th>
<th>(k = 3)</th>
<th>(k \geq 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undirected</td>
<td>(O(n))</td>
<td>(O(n))</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td></td>
<td>[BK79]</td>
<td>[HN14]</td>
<td>[ALN15]</td>
<td>[ALN15]</td>
</tr>
<tr>
<td>Upward</td>
<td>(O(n))</td>
<td>(OPEN)</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td></td>
<td>[HP99]</td>
<td></td>
<td>[Theorem 1]</td>
<td>[Theorem 1]</td>
</tr>
<tr>
<td>Matching</td>
<td>(O(n))</td>
<td>(O(n))</td>
<td>(OPEN)</td>
<td>NP-complete</td>
</tr>
<tr>
<td></td>
<td>[HP99]</td>
<td>[Theorem 3]</td>
<td></td>
<td>[Theorem 2]</td>
</tr>
</tbody>
</table>
Connections to Origami

- $k$ page Matching motivated by map folding problem
- **Map folding problem**: Given an $m \times n$ grid pattern with specified mountains and valleys, find a flat folded state
- Reduces to Matching
Connections to Origami

- \( k \) page Matching motivated by map folding problem
- **Map folding problem**: Given an \( m \times n \) grid pattern with specified mountains and valleys, find a flat folded state
- Reduces to Matching
- \( 1 \times n \rightarrow 2 \) page Matching (linear time) and \( 2 \times n \rightarrow 3 \) page Matching
- \( 2 \times n \) map folding currently has complicated \( O(n^9) \) algorithm
- **Open Question**: Can we get a better algorithm for \( 2 \times n \) map folding via reduction to 3 page Matching?
3 page Upward is NP-complete

- Reduction from NP-complete problem BETWEENNESS
- Betweenness:
  - Given $L$ variables and $C$ clauses where $\langle a, b, c \rangle$ is a clause
  - Find total ordering $\phi$ such that $\phi(a) < \phi(b) < \phi(c)$ or $\phi(c) < \phi(b) < \phi(a)$ is true for all clauses
- Given instance $(L, C)$ of betweenness construct instance of 3 page Matching $(G, P)$ where $P = \{RED, BLUE, GREEN\}$
- A solution $\pi$ to $(G, P)$ corresponds with a solution $\phi$ to $(L, C)$
Fig. 6: Example of a full construction of a reduction. Here, the instance of *Betweenness* is $(L, C)$ where $L = \{a, b, c, d\}$ and $C = \{\langle d, b, a \rangle, \langle b, c, d \rangle, \langle a, b, c \rangle\}$.
Ordered Triple Gadgets – used to represent clauses

Full construction of \((G, P)\) from \((L, C)\) where \(L = \{a, b, c, d\}\) and \(C = \{(d, b, a), (b, c, d), (a, b, c)\}\)
Order preserving gadgets – used to represent ordering of variables

Full construction of \((G, P)\) from \((L, C)\) where \(L = \{a, b, c, d\}\) and \(C = \{(d, b, a), (b, c, d), (a, b, c)\}\)
Ordering $\phi = [a, b, c, d]$

Full construction of $(G, P)$ from $(L, C)$
where $L = \{a, b, c, d\}$ and

$C = \{(d, b, a), (b, c, d), (a, b, c)\}$
Ordered Triple Gadgets

**Ordered triple gadget for clause** \(\langle a, b, c \rangle\)

**Two ways of embedding ordered triple gadget**
Ordered Triple Gadgets

**Ordered triple gadget for clause** \( \langle a, b, c \rangle \)

**Two ways of embedding ordered triple gadget**

\( b \) always stays between \( a \) and \( c \)

Definition 2

Refer to Fig. 1 for an example construction.

Ordered Triple Gadgets

Fig. 1: Ordered triple gadget (left) and order preserving gadget for odd integer \( i \) between and for even integer \( i \) directly to \( i \).

Fig. 2: The two possible embeddings of the ordered triple gadget.
Ordered Triple Gadgets

Fig. 2: The two possible embeddings of the ordered triple gadget.

Red edges connect to order preserving gadgets

Ordered triple gadget for clause \( \langle a, b, c \rangle \)

Two ways of embedding ordered triple gadget

Let \( \alpha_1 \) be an instance of \( L, C \) arbitrarily. For the triple \( \alpha_i \) are respectively connected to \( l_i \), \( a_i \), \( b_i \), \( c_i \), and \( h_i \). Then, create directed edges as we prove in Lemmas 1 and 2.

Let \( \alpha_1 \) be an instance of \( L, C \) arbitrarily. For the triple \( \alpha_i \) are respectively connected to \( l_i \), \( a_i \), \( b_i \), \( c_i \), and \( h_i \). Then, create directed edges as we prove in Lemmas 1 and 2.

Ordered triple gadget for clause \( \langle a, b, c \rangle \)

Two ways of embedding ordered triple gadget
Order Preserving Gadgets

Refer to Fig. 1 for an example construction. Nodes $i \leq h \leq i + 1$ are arbitrarily directed to $(\text{odd elements})$. Order the set $\{0, 2, \ldots, h\}$ arbitrarily. For the order preserving gadget (which is all ordered the same and are the reverse element vertices of $d_i$), the red edge from $a_i$ to $b_i$ is encoded in the choice $(\text{where } b_i \neq c_i)$. By construction, the paths from $a_i$ to $b_i$ are between element vertices and vertices in ordered triple gadgets of the same color between element vertices and vertices in ordered triple gadgets. By these lemmas, if $(L, C) \in \mathbb{R}$, we can obtain a valid ordering of all element vertices of even order preserving gadgets. Betweenness $(\text{element vertices})$ represents a path whose vertices appear subsequently in the order of all even order preserving gadgets. The topological order of $G, P$ is set so that no two edges of the paths from $a_i$ to $b_i$ cross. A red and a green edge is used to “go through” each dashed arrow to the path’s endpoint. Such an embedding allows the alternating red/green paths to be realized so that no crossing occur with the three red edges coming from the path's endpoint. All element vertices should occur in the reverse order of the $L, C$ order of all even order preserving gadgets. The paths from $a_i$ to $b_i$ represents a path whose vertices appear subsequently in the order of all even order preserving gadgets (which are all ordered the same and are the reverse element vertices of $d_i$).

Order Preserving Gadgets

Odd index order preserving gadget

Refer to Fig. 1 for an example construction. Nodes $i \leq h \leq i + 1$ are arbitrarily directed to $(\text{odd elements})$. Order the set $\{0, 2, \ldots, h\}$ arbitrarily. For the order preserving gadget (which is all ordered the same and are the reverse element vertices of $d_i$), the red edge from $a_i$ to $b_i$ is encoded in the choice $(\text{where } b_i \neq c_i)$. By construction, the paths from $a_i$ to $b_i$ are between element vertices and vertices in ordered triple gadgets of the same color between element vertices and vertices in ordered triple gadgets. By these lemmas, if $(L, C) \in \mathbb{R}$, we can obtain a valid ordering of all element vertices of even order preserving gadgets. Betweenness $(\text{element vertices})$ represents a path whose vertices appear subsequently in the order of all even order preserving gadgets. The topological order of $G, P$ is set so that no two edges of the paths from $a_i$ to $b_i$ cross. A red and a green edge is used to “go through” each dashed arrow to the path’s endpoint. Such an embedding allows the alternating red/green paths to be realized so that no crossing occur with the three red edges coming from the path's endpoint. All element vertices should occur in the reverse order of the $L, C$ order of all even order preserving gadgets. The paths from $a_i$ to $b_i$ represents a path whose vertices appear subsequently in the order of all even order preserving gadgets (which are all ordered the same and are the reverse element vertices of $d_i$).
**Order Preserving Gadgets**

Odd index order preserving gadget

Even index order preserving gadget

Refer to Figure 1 for an example construction. Nodes $0, 1, \ldots, n$ are the element vertices of an ordered triple gadget. The red edge from $a_i \mapsto b_i, c_i$ and blue edge from $b_i \mapsto c_i$ ensure that the path's endpoint. Such an embedding allows the alternating red/green paths to cross. A red and a green edge is used to “go through” each dash arrow to a point. The paths from $a, b, c \mapsto i$ and $i \mapsto a, b, c$ ensure that the conditions in Lemmas 1, 2, 3 and Corollary 1 are always met. For even $0, 1, \ldots, n$ and latest nodes in the gadget are respectively connected to $l, h$ where $l$ is an odd integer between and including $h$ and $l$. Order the set $0, 1, \ldots, n$ as we prove in Lemmas 1 and 2. For the odd order preserving gadgets (which are all ordered the same and are the reverse (see Figure 6) since all ordered triple gadgets are satisfied by the ordering of the element vertices of $d_i$. Let $0, 1, \ldots, n$ be an instance of $(L, C, P)$. The topological order of $m, n$ ensures that the conditions in Lemmas 1, 2, 3 and Corollary 1 are always met. In our reduction in Section 3.2 is a positive instance of $(L, C, P)$. The topological order of $m, n$ ensures that the conditions in Lemmas 1, 2, 3 and Corollary 1 are always met.
Order Preserving Gadgets

Blue edges go from even index $j$ to odd index $j - 1$
Order Preserving Gadgets

Green edges go from even index $j$ to odd index $j + 1$
between

Definition 2

Refer to Fig. 1 for an example construction. Nodes

Fig. 2: The two possible embeddings of the ordered triple gadget.

Ordered Triple Gadget.

Order Preserving Gadgets

Betweenness

Valid embedding of order preserving gadget

UPBE-
Order Preserving Gadgets

Red edges from ordered triple gadget ensures valid ordering of variables in clauses
Order Preserving Gadgets

Red edges from ordered triple gadget ensures valid ordering of variables in clauses and corresponding vertices in the order preserving gadget
Order Preserving Gadgets

Order of $a_i, b_i, c_i, d_i$ in $\pi$ represent valid ordering of $a, b, c, d$ in $\phi$
Fig. 6: Example of a full construction of a reduction. Here, the instance of \( \text{Be-betweenness} \) is \((L, C)\) where \(L = \{a, b, c, d\}\) and \(C = \{h_a, b, c;i, h_b, c, d;i, h_d, b, a;i\}\). Full construction maintains order of ordered triple gadgets and order preserving gadgets.
4 page Matching is NP-complete

Fig. 3: Gadgets for the reduction to UMPBE-4.

For odd $i$ in $\{1, \ldots, 2^m\}$ we connect gadgets with yellow edges $(h_i, r_i)$ and $(r_i, l_i)$ if $i > 1$, and with the red edge $(r_i+1, l_i)$ if $i < 2^m$. Lemmas 1 holds for the new gadget replacing $x_{00}$ by $x_i$. We omit its proof due to the similarity. The dashed arrows in Figure 3 represent paths of alternating colors as described in the next paragraph. Lemma 3 also trivially holds. Therefore, given a valid order $\pi$ of vertices of $G$, the order of element vertices corresponds to a solution of the Betweenness instance.

It remains to show that, given a solution of the Betweenness instance, we can obtain a solution $\pi$ for the produced instance. The order in which the gadgets are embedded are the same as in the proof of Theorem 1 and, therefore, no edge between gadgets cross. We now show that each gadget has a cross-free embedding using $\pi$. The embedding of the ordered triple gadget is very similar to that shown in Figure 2 and we chose $\pi(x_i) > \pi(r_i)$ or $\pi(x_i) < \pi(r_i)$ depending on whether $a$ appears before $c$ or vice-versa in $x$. In the order preserving gadget, we use the same order (resp., reverse order) of for even (resp., odd) $j$. Notice that the order preserving gadget now contains a binary tree and we cannot chose arbitrarily the order of vertices $x_{\cdot j}$ for $x \in L$. We call this tree the binary order preserving tree that ensures that $\pi(x_i) > \pi(r_i)$ and $\pi(x) < \pi(r)$ for all $x \in L$ for odd $i$ and even $j$. The tree is obtained by ordering the vertices $x_{\cdot j}$ arbitrarily and building a binary tree that alternates between green/blue.
4 page Matching is NP-complete

Fig. 3: Gadgets for the reduction to UMPBE-4.

The partitions in $P = \{\text{Red}, \text{Blue}, \text{Green}, \text{Yellow}\}$ as colors. The gadgets adapted from Section 3.1 are shown in Figure 3.

For odd $i$ in $\{1, \ldots, 2m\}$ we connect gadgets with yellow edges ($h_i, r_i$) and ($r_{i+1}, l_i$) if $i > 1$, and with the red edge ($r_i, l_i$) if $i < 2m_1$.

Lemma 1 holds for the new gadget replacing $x_0$ by $x_i$. We omit its proof due to the similarity. The dashed arrows in Figure 3 represent paths of alternating colors as described in the next paragraph. Lemma 3 also trivially holds. Therefore, given a valid order $\pi$ of vertices of $G$, the order of element vertices corresponds to a solution of the Betweenness instance.

It remains to show that, given a solution of the Betweenness instance, we can obtain a solution $\pi$ for the produced instance. The order in which the gadgets are embedded are the same as in the proof of Theorem 1 and, therefore, no edge between gadgets cross. We now show that each gadget has a cross-free embedding using $\mathcal{L}$. The embedding of the ordered triple gadget is very similar to that shown in Figure 2 and we chose $\pi(x_i) > \pi(r_i)$ or $\pi(x_i) < \pi(r_i)$ depending on whether $a$ appears before $c$ or vice-versa in $x_i$.

In the order preserving gadget, we use the same order (resp., reverse order) of $x_j$ for even (resp., odd $j$).

Ordered triple gadget  Odd index order preserving gadget

Fig. 4: Embedding the order preserving gadget in Figure 3 (center).
UMPBE-4 is NP-complete

Fig. 3: Gadgets for the reduction to UMPBE-4.

The partitions in $P = \{\text{Red}, \text{Blue}, \text{Green}, \text{Yellow}\}$ as colors. The gadgets adapted from Section 3.1 are shown in Figure 3.

For odd $i$ in $\{1, \ldots, 2^m - 1\}$ we connect gadgets with yellow edges ($(h_i, r_i)$) and ($(r_i, l_i)$) if $i > 1$, and with the red edge ($(r_i + 1, l_i)$) if $i < 2^m - 1$.

Lemmas 1 holds for the new gadget replacing $x_{i0}$ by $x_{i1}$. We omit its proof due to the similarity. The dashed arrows in Figure 3 represent paths of alternating colors as described in the next paragraph. Lemma 3 also trivially holds. Therefore, given a valid order $\Pi$ of vertices of $G$, the order of element vertices corresponds to a solution of the Betweenness instance.

It remains to show that, given a solution of the Betweenness instance, we can obtain a solution $\Pi$ for the produced instance. The order in which the gadgets are embedded are the same as in the proof of Theorem 1 and, therefore, no edge between gadgets cross. We now show that each gadget has a cross-free embedding using $G$. The embedding of the ordered triple gadget is very similar to that shown in Figure 2 and we chose $\Pi((i, k)) > \Pi((j, k))$ or $\Pi((i, k)) < \Pi((j, k))$ depending on whether $a$ appears before $c$ or vice-versa in $G$. In the order preserving gadget, we use the same order (resp., reverse order) of $x_j$ for even (resp., odd) $j$.

Notice that the order preserving gadget now contains a binary tree and we cannot choose arbitrarily the order of vertices $x_{i1}$ for $x_i \in L$. We call this tree the binary order preserving tree that ensures that $\Pi(x_i) > \Pi(r_i)$ and $\Pi(x_j) < \Pi(r_j)$ for all $x_i \in L$ for odd $i$ and even $j$. The tree is obtained by ordering the vertices $x_{i1}$ arbitrarily and building a binary tree that alternates between green/blue.
4 page Matching is NP-complete

Order preserving trees to preserve order of ordered triple and order preserving gadgets
4 page Matching is NP-complete

Fig. 3: Gadgets for the reduction to UMPBE-4.

For odd $i$ in $\{1, \ldots, 2m\}$ we connect gadgets with yellow edges ($h_i, r_i$), and with the red edge ($r_{i+1}, l_i$) if $i > 1$, and with the blue edge ($r_i, l_{i-1}$) if $i < 2m - 1$.

Lemmas 1 holds for the new gadget replacing $x_0$ by $x_i$. We omit its proof due to the similarity. The dashed arrows in Figure 3 represent paths of alternating colors as described in the next paragraph. Lemma 3 also trivially holds. Therefore, given a valid order $\pi$ of vertices of $G$, the order of element vertices corresponds to a solution of the Betweenness instance.

It remains to show that, given a solution of the Betweenness instance, we can obtain a solution $\pi$ for the produced instance. The order in which the gadgets are embedded are the same as in the proof of Theorem 1 and, therefore, no edge between gadgets cross. We now show that each gadget has a cross-free embedding using $\pi$. The embedding of the ordered triple gadget is very similar to that shown in Figure 2 and we chose $\pi$ depending on whether $a$ appears before $c$ or vice-versa in $\{x\}$. In the order preserving gadget, we use the same order (resp., reverse order) of $\pi$ for even (resp., odd) $j$.

Dashed red lines represent paths of alternating colors of length $n$. The binary order preserving gadget now contains a binary tree and we cannot chose arbitrarily the order of vertices $x_i$ for $x \in L$. We call this tree the binary order preserving tree that ensures that $\pi(x_i) > \pi(r_i)$ and $\pi(x_j) < \pi(r_j)$ for all $x \in L$ for odd $i$ and even $j$.

The tree is obtained by ordering the vertices $x_i$ arbitrarily and building a binary tree that alternates between green/blue.
It remains to show that, given a solution to a solution given a valid order as described in the next paragraph. Lemma 3 also trivially holds. Therefore, holds for the new gadget replacing the partitions in a partition.

Betweenness gadget (红色边) and even (蓝色边) for odd (绿色边). The tree is obtained by ordering the vertices of vertices of G, the order of element vertices corresponds to the order of vertices of G. The embedding of the ordered triple gadget is very similar to the embedding paths of the gadget for the reduction to the problem of finding a betweenness gadget. The dashed arrows in Figure 3 represent paths of alternating colors. The gadgets adapted as colors. The gadgets adapted in the order preserving gadget, the order of element vertices corresponds to the order of vertices of G, the order of element vertices corresponds to the order of vertices of G. The similar gadget now contains a binary tree and we cannot choose arbitrarily the order of vertices that appears before or after another.

Notice that the order preserving gadget now contains a binary tree and we cannot choose arbitrarily the order of vertices that appears before or after another.

Vertices are inserted in insertion sort manner.
Embedding paths

Vertices are inserted in insertion sort manner
For all binary order preserving trees, we cannot choose the order of vertices arbitrarily and building a binary tree that alternates between green/blue holds for even (resp., odd) vertices. The tree is obtained by ordering the vertices as colors. The gadgets adapted to the partitions in a gadget are the same as in the proof of Theorem 1 and, therefore, we can obtain a solution to the problem as described in the next paragraph. Lemma 3 also trivially holds. Therefore, the order preserving gadget now contains a binary tree and we notice that the order of element vertices corresponds to the order in which the gadgets are embedded.

If we connect gadgets with yellow edges, we call this tree the tree obtained by ordering the vertices given a valid order. The dashed arrows in Figure 3 represent paths of alternating colors. The similarity holds for the new gadget replacing the one from Section 3.1 as shown in Figure 3.

Fig. 4: Embedding the order preserving gadget in Figure 3 (center).

Vertices are inserted in insertion sort manner.
Linear Time 2 page Matching

- Linear time algorithm for 2 page Matching by reducing to 1D origami or single vertex flat foldability

(a) 1D Origami
(b) Single vertex foldability
Linear Time 2 page Matching

- Linear time algorithm for 2 page Matching by reducing to 1D origami or single vertex flat foldability

For both the case of the path and the cycle, we create an instance of 1D origami and single vertex flat foldability, respectively, in the following way. For each edge $e \in E$, we create a mountain crease if $e \in R$ and a valley crease if $e \in B$. Each face of the produced instance represents a vertex in $G$. The reduction will thus produce an instance where each face of the origami has the same length, which can be viewed as a linkage formed by identical bars. If $G$ is a path (resp., cycle), the output will be a list (resp., circular list) containing the assignment of the creases on a line segment (resp., a single vertex origami).

Start with one endpoint of the path or with an arbitrary vertex of the cycle. Traverse the undirected version of $G$ using BFS. For each edge traversed add mountain (resp., valley) to the end of the list if the traversed edge corresponds to an edge in $E_1$ with the same direction of the traversal or to an edge in $E_2$ in the opposite direction (resp., corresponds to an edge in $E_2$ with the same direction of the traversal or to an edge in $E_1$ in the opposite direction). Since every edge is traversed once, the size of the list is $n$ (resp., $n$ if a cycle).

Thus, the only difference between single vertex crease patterns and 1D origami is that the faces form a cycle as opposed to a line segment, respectively. Due to the similarity of the reduction models for paths and cycles, it suffices to show the equivalence between instances when $G$ is a path. As previously stated, each face of the paper corresponds to a vertex in $G$. Each crease represents an edge in $G$ and whether the crease is a mountain fold or a valley fold in the final state of the origami determines the partition of the edges of $G$ into $R$ or $B$ edges. If we consider the starting vertex as the leftmost face of the unfolded paper and that $f_1$ is not flipped in the folded state, a mountain fold puts the adjacent face $f_2$ below $f_1$. Without loss of generality, the edge in $G$ that represents the connection between $f_1$ and $f_2$ is in $E_2$ and points from $f_2$ to $f_1$. By repeating the argument for every edge, we conclude that $G$ represents the above/below relation of faces of the folded state of the 1D origami and $E_1$ (resp., $E_2$) represents the creases that lie right (resp., left) of the folded state (see Figure 5). Then, it is easy to verify that the origami is flat-foldable if $(G, \{E_1, E_2\})$ is a positive instance of UMPBE.

Fig. 5: A 1D origami crease pattern is shown (left) with mountain/valley labeled as M/V respectively, together with its folded state (center) and the corresponding UMPBE-2 instance (right).