The Poset Cover Problem

Lenwood S. Heath¹, Ajit Kumar Nema²
¹Department of Computer Science, Virginia Tech, Blacksburg, USA
²Core Quantative Strategies, Goldman Sachs, India Pvt. Ltd., Bangalore, India
Email: heath@vt.edu, ajitnema@gmail.com

Received March 1, 2013; revised April 1, 2013; accepted May 15, 2013

ABSTRACT

A partial order or poset \( P = (X, \prec) \) on a (finite) base set \( X \) determines the set \( \mathcal{L}(P) \) of linear extensions of \( P \). The problem of computing, for a poset \( P \), the cardinality of \( \mathcal{L}(P) \) is \#P-complete. A set \( \{P_1, P_2, \ldots, P_k\} \) of posets on \( X \) covers the set of linear orders that is the union of the \( \mathcal{L}(P_i) \). Given linear orders \( L_1, L_2, \ldots, L_m \) on \( X \), the Poset Cover problem is to determine the smallest number of posets that cover \( \{L_1, L_2, \ldots, L_m\} \). Here, we show that the decision version of this problem is \( NP \)-complete. On the positive side, we explore the use of cover relations for finding posets that cover a set of linear orders and present a polynomial-time algorithm to find a partial poset cover.

Keywords: Linear Orders; Partial Orders; \( NP \)-Completeness; Algorithms

1. Introduction

Finite partial orders or posets have numerous applications, including scheduling [1-8], molecular evolution [9-12], data mining [13-17], graph theory [18-23], and algebra [24-27]. Many applications implicitly or explicitly involve linear extensions of posets. For example, the solution of many scheduling problems requires a linearization of the jobs being scheduled consistent with some precedence constraints given by a poset. As the number of linear extensions of a poset may be exponential in the number of elements of the base set, many computational problems related to linear extensions are not solvable in polynomial time. Ruskey [28], West [29], Pruesse and Ruskey [30], Canfield and Williamson [31], Korsh and LaFollette [32], and Ono and Nakano [33] provide algorithms to generate all of the linear extensions of a finite poset. As the size of a solution may be exponentially large, these algorithms emphasize the ability to generate each successive linear extension in polynomial time, at least on average. Sampling the linear extensions of a poset is easier. Bubley and Dyer [34] use a rapidly mixing Markov chain to generate a random linear extension of a finite poset, sampled almost uniformly.

Problems in mining order information from databases of sequences (see, e.g., [16,17,35,36]) have an inverted character from that of many computational problems involving posets. Here, a problem instance is a set of linear orders of items from some universal set, while a solution is one or more posets that well explain, through their linear extensions, a significant number of the linear orders. An example from computational neuroscience [37] might go as follows. Each item is the firing of a neuron, while each linear order is a sequence of neuronal firings, ordered in time from an experiment. The solution is a neural circuit that explains a set of such linear orders. These novel problems are ripe for mathematical formalization and study. In this paper, we define and study one such problem. A problem instance is a set of permutations of a base set, and a solution covers the instance with linear extensions (Section 2). We prove that the Poset Cover problem (a decision problem) is \( NP \)-complete in Section 3. In Section 4, we explore how cover relations relate to poset covers. Finally, we develop a polynomial-time algorithm to find a partial cover in Section 5.

2. Preliminaries

In this section, we establish terminology and notation and prove some basic results.

A partial order or poset \( P \) is an irreflexive, asymmetric, and transitive binary relation \( \prec \) defined on a finite set \( X \) of cardinality \( n \geq 1 \). We write \( P \) as the
ordered pair $P = (X, \prec_r)$. Equivalently, poset $P$ is a transitive directed acyclic graph (DAG), namely,

$$P = \{ (x, y) \mid x \prec_r y \}.$$ If $G$ is a DAG, then its transitive closure is a poset by this equivalence. The rank function $ho : X \to \{1, 2, \ldots, n\}$ is given by

$$\rho(x) = 1 + \max\{y \mid y \prec_r x\}.$$ The empty poset is $\epsilon = (X, \emptyset).$

Let $x, y \in X$ be distinct. Then $x$ and $y$ are comparable in $P$, written $x \ll y$, if $x \prec y$ or $y \prec x$, while $x$ and $y$ are incomparable, written $x \not\ll y$, otherwise. Moreover, $x$ is covered by $y$ or $y$ covers $x$, written $x \ll y$, if $x \prec y$ and there is no $z \in X$ such that $x \prec z \ll y$. In this case, the ordered pair $(x, y)$ is a cover relation for $P$. It is well-known that a (finite) poset is uniquely determined by its set of cover relations (see [38]).

If $P_1 = (X, \ll_1)$ and $P_2 = (X, \ll_2)$ are posets on the same set $X$, then $P_2 \subseteq P_1$, if, for all $a \in P_2$, $a \ll_1 b$, $a \ll_2 b$ implies $a \ll_2 b$. The relation $\subseteq$ is reflexive, antisymmetric, and transitive.

A linear order $L = (X, \prec_1)$ on a poset $X$ such that, for $x, y \in X$, either $x = y$ or $x \ll_1 y$ holds. If $L$ is a linear order, then the rank function $\rho : X \to \{1, 2, \ldots, n\}$ is a bijection. Setting $x = \rho^{-1}(i)$, $L$ can be written as the sequence $L = x_1, x_2, \ldots, x_n$, which is a permutation of $X$. Also, we write $L[i]$ for the element of rank $i$ in $L$. A linear extension $L$ of a poset $P$ is a linear order such that $P \subseteq L$. The set of all linear extensions of $P$ is $\mathcal{L}(P)$. Note that $\mathcal{L}(\epsilon)$ is the set of all linear orders on $X$. Brightwell and Winkler [39] prove that the problem of determining $|\mathcal{L}(P)|$ for a poset $P$ is $\#P$-complete.

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ be a set of $k$ posets on $X$. This set covers the set of linear orders $Y = \bigcup_{P \in \mathcal{P}} \mathcal{L}(P)$.

A poset $P \in \mathcal{P}$ is maximal in $Y$ if $\mathcal{L}(P) \subseteq Y$ and there is no poset $P' \supset P$ of $X$ such that $P \neq P', P' \subseteq P$, and $\mathcal{L}(P') \subseteq Y$. Let $\mathcal{P}$ be a set of posets on $X$, and let $Y$ be a set of linear orders on $X$. $\mathcal{P}$ blankets $Y$ if $Y \subseteq \bigcup_{P \in \mathcal{P}} \mathcal{L}(P)$.

**Lemma 1** Let $Y$ be the set of linear orders that is covered by a set $\mathcal{P}$ of posets of cardinality $k$. Then, there exists a cover $\hat{\mathcal{P}}$ of cardinality $k' \leq k$ that also covers $Y$ such that every poset in $\hat{\mathcal{P}}$ is maximal in $Y$.

**Proof.** We construct $\hat{\mathcal{P}}$ by examining each poset in $\mathcal{P}$. Let $P \in \mathcal{P}$. If $P$ is maximal in $Y$, then add $P$ to $\hat{\mathcal{P}}$. Otherwise, let $P'$ be a poset of minimum cardinality (as a set of ordered pairs) such that $P' \subseteq P$ and $\mathcal{L}(P') \subseteq Y$. Since $P$ is not maximal, $P \neq P'$. Moreover, any poset $P''$ contained in $P'$ of smaller cardinality will have $\mathcal{L}(P'') \not\subseteq Y$. Add $P'$ to $\hat{\mathcal{P}}$.

The constructed $\hat{\mathcal{P}}$ has cardinality $\leq k$. Moreover, $\hat{\mathcal{P}}$ also covers $Y$ and every poset in $\hat{\mathcal{P}}$ is maximal in $Y$. The lemma follows.

In this paper, we are interested in reversing the cover relationship by addressing the problem of finding a minimum set of posets that covers a given set of linear orders. As a decision problem, this is

**Poset Cover**

**INSTANCE:** A base set $X$ of cardinality $n \geq 1$; a nonempty set $Y = \{L_1, L_2, \ldots, L_m\}$ of linear orders over $X$; and an integer $K \leq m$.

**QUESTION:** Is there a set $\mathcal{P}$ of posets on $X$ of cardinality $\leq K$ that covers $Y$?

This problem is shown to be NP-complete in Section 3.

Let $L = x_1, x_2, \ldots, x_n$ be a linear order on $X$. For each $i$ satisfying $1 \leq i \leq n-1$, the $i$-swap of $L$ is the linear order $\text{Swap}(i; L) = x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n$. Let $L' = \text{Swap}(i; L)$. Evidently, $\text{Swap}(i; L') = \text{Swap}(i, L')$.

For pairs $(L, L')$ that are $i$-swaps of each other, for some $i$, we define the function $\text{SwapIndex}(i; L, L') = i$. Note that the set differences of $L$ and $L'$, namely $L \setminus L' = \{x_i \mid i \in L, i \notin L'\}$ and $L' \setminus L = \{x_i \mid i \in L', i \notin L\}$, each consist of a single ordered pair. In this case, the swap pair for $L$ and $L'$ is the unordered pair $\text{SwapPair}(L, L') = \{x_i, x_{i+1}\}$; otherwise, $\text{SwapPair}(L, L') = \emptyset$. Two linear orders $L_1$ and $L_2$ differ by a swap, written $L_1 \leftrightarrow L_2$, if $L_1 \leftrightarrow L_2$, for some $i$. Since $L_1 \leftrightarrow L_2$ if and only if $L_2 \leftrightarrow L_1$, the $\leftrightarrow$ relation is also symmetric. If $L_1 \leftrightarrow L_2$, $a \ll_1 b$, and $b \ll_2 a$, then we write $\text{Swap}(i; L_1 \leftrightarrow L_2)$ to mean that $L_1 \leftrightarrow L_2$ for some $i$, where the elements swapped are $a$ and $b$.

Let $Y$ be a set of linear orders on $X$. The swap graph of $Y$ is the undirected graph $G(Y) = \{(L, L') \mid L \leftrightarrow L'\}$. An edge $(L, L')$ of $G(Y)$ is labeled $\text{SwapPair}(L, L')$. Let $P$ be a poset on $X$, and let $L \in Y$. Then, $P'$ is a partial cover of $Y$ including $L$ if $L \in \mathcal{L}(P)$ and $\mathcal{L}(P') \subseteq Y$. The swap graph is the same as the adjacent transposition graph of Pruesse and Ruskey [30]. The swap graph of $Y$ is bipartite, since every edge connects an even permutation to an odd permutation. Moreover, the swap graph $G(Y)$ of the linear extensions of a single poset is connected (see [30]).

Let $\Pi$ be the set of all posets on $X$. Let $A \subseteq X \times X$ be a set of ordered pairs. The up-set of $A$ is
Up(A) = \{ P \in \Pi | a <_P b \text{ for all } (a,b) \in A \}.

Up(A) is empty if and only if the directed graph (X, A) contains cycles. Let
B \subseteq \{(a,b) | a, b \in X \text{ and } a \neq b\} be a set of unordered pairs. The down-set of B is
Down(B) = \{ P \in \Pi | a \preceq_P b \text{ for all } (a,b) \in B \}.

Down(\emptyset) = \Pi,\text{ and we always have the empty poset } e \in Down(B).
If Up(A) \neq \emptyset, then define the minimal element in Up(A) to be
Min(A) = \bigcap_{P \in Up(a)} P.

The following properties of Min(A) follow directly from the definitions.

Lemma 2 A \subseteq Min(A) and Up(\{\}) = \{ P | Min(A) \subseteq P \}.

We have the following properties of up-sets and down-sets.

Lemma 3 Let A, B \subseteq X \times X. If A \subseteq B, then Up(B) \subseteq Up(A). Let
C, D \subseteq \{(c,d) | c, d \in X \text{ and } c \neq d\}. If C \subseteq D, then
Down(D) \subseteq Down(C).

Proof. Suppose that A \subseteq B. By the definition of up-sets,
Up(B) = \{ P \in \Pi | a <_P b \text{ for all } (a,b) \in B \}
\subseteq \{ P \in \Pi | a <_P b \text{ for all } (a,b) \in A \} = Up(A).

Now, suppose that C \subseteq D. Then,
Down(D) = \{ P \in \Pi | a \preceq_P b \text{ for all } (a,b) \in D \}
\subseteq \{ P \in \Pi | a \preceq_P b \text{ for all } (a,b) \in C \} = Down(C),
by the definition of down-sets.

3. NP-Completeness of Poset Cover

In this section, we show that PosetCover is NP-complete, in the process using the following known NP-complete decision problem [40].

Cubic Vertex Cover

INSTANCE: A nonempty undirected graph G = (V, E) that is cubic, that is, in which every vertex has degree 3; and an integer K \leq |V|.

QUESTION: Is there a subset V' \subseteq V of cardinality \leq K such that every edge in E is incident on at least one vertex in V'?

Theorem 4 Poset Cover is NP-complete.

Proof. We show that Poset Cover is in NP and that Cubic Vertex Cover reduces to Poset Cover in polynomial time.

We first show that Poset Cover is in NP. Let X, Y = \{L_1, L_2, \ldots, L_m\}, and K constitute an instance of Poset Cover, and let P = \{P_1, P_2, \ldots, P_k\} be a set of posets on X. First, it is easy to check whether k \leq K in time polynomial in n and m; if k > K, then return No. Second, if the cardinality check succeeds, check whether P covers Y as follows. For each poset P_i in turn, use the Korsh and LaFollette [32] algorithm to generate all the linear extensions of P_i, one at a time, in constant time per linear extension. As each linear extension L \in L(P_i) is generated, check whether L \in Y. If not, then return No. If so, then mark that element of Y Covered. Note that the number of linear orders generated by a run of the Korsh and LaFollette algorithm before completion or returning No is at most m. Hence, the worst-case time for one run of the algorithm, including the checking, is O(mn). Once all the posets and their linear extensions are processed, check whether every element of Y is marked Covered. If so, then return Yes; otherwise, return No. We find that the worst-case time to check whether P covers Y is O(m + m^2 n), since k \leq K \leq m. This is polynomial in the size of the original instance. We conclude that Poset Cover is in NP.

Now, let G = (V, E) and K constitute an instance of Cubic Vertex Cover. Without loss of generality, assume that |V| = \ell and that V = \{1, 2, \ldots, \ell\}. Let s = |E| = 3\ell/2, and let E = \{e_1, e_2, \ldots, e_s\} be an arbitrary labeling of the s edges of G. As a running example of our reduction, we provide the cubic graph in Figure 1, with \ell = 6 vertices and s = 9 edges. To complete the instance of Cubic Vertex Cover, set K = 4.

Let n = 2(s + 2), and let X = \{x_1, x_2, \ldots, x_n\} be a base set of n elements. Let L_0, the base order, be the linear order on X specified by
x_1 \leq L_0 x_2 \leq L_0 \cdots \leq L_0 x_n.

We view the elements of X as consisting of s + 2 adjacent, non-overlapping pairs. Specifically, the pairs are x_{2i-1} and x_{2i}, where 1 \leq i \leq s + 2. All elements of Y are obtained by a small set of swaps of such pairs, applied to L_0.

Figure 1. A cubic graph as part of an instance of cubic vertex cover.
The first \( s \) \( s \) pairs correspond to the \( s \) edges in a natural way. In particular, edge \( e_i \in E \) is associated with the edge order \( L_0 = \text{Swap}[L_0; 2i-1] \). Continuing the example, we set \( n = 2(s + 2) = 22 \),

\[
X = \{x_1, x_2, \ldots, x_{22}\}, \quad L_0 = x_1, x_2, \ldots, x_{22},
\]

and, for example,

\[
L_0 = x_2, x_1, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}.
\]

For each vertex \( \nu \in V \), there are three edges incident on \( \nu \); let the indices of those edges be \( \chi[\nu, 1], \chi[\nu, 2], \) and \( \chi[\nu, 3] \). For each pair \( e_{[\nu, 1]} \) and \( e_{[\nu, 2]} \) of these edges, we define the pair edge order to be

\[
L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*}[\nu, 1] = \text{Swap} \{L_0; 2\chi[\nu, i]-1; 2\chi[\nu, j]-1\}.
\]

For the running example, there are 18 pair edge orders. For each triple \( e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^* \), we define the triple edge order to be

\[
L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*}[\nu, 1] = \text{Swap} \{L_0; 2\chi[\nu, i]-1; 2\chi[\nu, j]-1; 2\chi[\nu, k]-1\}.
\]

For the running example, there are 6 triple edge orders. The primary orders are the base, edge, pair edge, and triple edge orders. For primary pair edge order \( L_{e_{[\nu, 1]}^*} \), there is a corresponding secondary pair edge order obtained by swapping \( x_{2s+1} \) and \( x_{2s+2} \), which is

\[
L'_{e_{[\nu, 1]}^*} = \text{Swap} \{L_{e_{[\nu, 1]}^*}; 2s + 1\}.
\]

For primary triple edge order \( L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*} \), there is a corresponding secondary triple edge order obtained by swapping \( x_{2s+3} \) and \( x_{2s+4} \), which is

\[
L'_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*} = \text{Swap} \{L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*}; 2s + 3\}.
\]

For the running example, there are 18 secondary pair edge orders and 6 secondary triple edge orders. Collect the various orders into five sets, as follows:

\[
A = \{L_i; 1 \leq i \leq s\},
\]

\[
B = \{L_{e_{[\nu, 1]}^*}; e_i \text{ and } e_j \text{ are incident on some } \nu \in V\},
\]

\[
B' = \{L'_{e_{[\nu, 1]}^*}; e_i \text{ and } e_j \text{ are incident on some } \nu \in V\},
\]

\[
C = \{L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*}; \nu \in V\},
\]

\[
C' = \{L'_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*}; \nu \in V\}.
\]

We can now complete our instance of Poset Cover by setting

\[
Y = \{L_i \cup A \cup B \cup B' \cup C \cup C'\}
\]

and setting the integer parameter \( K' = K + 4\ell \). Note that \( |Y| = 1 + s + 8\ell \). For the running example, \( K' = 4 + 4 \times 6 = 28 \) and \( |Y| = 1 + 9 + 8 \times 6 = 58 \).

It remains to show that an instance of Cubic Vertex Cover is a Yes instance if and only if the corresponding instance of Poset Cover is a Yes instance.

Fix an instance \( G = (V, E) \) and \( K \) of Cubic Vertex Cover. Let \( X, Y, \) and \( K' \) constitute the corresponding instance of Poset Cover, as constructed above. By Lemma 1, we may assume that every element of a cover \( \mathcal{P} \) of \( Y \) is maximal in \( Y \). Since the elements of \( B' \) must be blanketed by any cover, we may assume that the set

\[
B' = \{L_{e_{[\nu, 1]}^*}(L_{e_{[\nu, 1]}^*}; e_i \text{ and } e_j \text{ incident on some } \nu \in V\}
\]

is a subset of \( \mathcal{P} \). Similarly, since the elements of \( C' \) must be blanketed by any cover, we may assume that the set

\[
C' = \{L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*}; \nu \in V\}
\]

is a subset of \( \mathcal{P} \). Note that \( |B' \cup C'| = 4\ell \) and that \( B' \cup C' \) blankets \( B \cup B' \cup C \).

First, assume that \( V' \in V \) is a vertex cover of \( G \) of cardinality at most \( K \). Define

\[
\mathcal{P} = B' \cup C' \cup \bigcup_{\nu \in V'} \{L_{e_{[\nu, 1]}^*}; e_i \text{ and } e_j \text{ incident on some } \nu \in V\}.
\]

Now, assume that \( \mathcal{P} \) is a cover of \( Y \) of cardinality at most \( K' \). By previous observations, it suffices to demonstrate that \( \mathcal{P} \) blankets \( L_i \cup A \cup B \cup B' \cup C \cup C' \) and that, by previous observations, \( |Y| = 4\ell + |V'| \leq 4\ell + K = K' \). Since \( \mathcal{P} \) covers \( Y \), \( \mathcal{D} \) must also cover \( L_i \cup A \cup B \cup B' \cup C \cup C' \).

Let \( e_i \in A \cup B \cup B' \cup C \cup C' \). Without loss of generality, we may assume that \( i \in [u, 1] \). There are two maximal posets that blanket \( L_i : L_i \cap L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*} \) and \( L_i \cap L_{e_{[\nu, 1]}^*, e_{[\nu, 2]}^*, e_{[\nu, 3]}^*} \). One of these posets must be in \( \mathcal{D} \).

Moreover, we may assume that \( \mathcal{D} \) contains only orders of this form, since each such order blankets \( L_i \), and the only other orders for \( \mathcal{D} \) to blanket are the \( L_i \)'s.
Define 

\[ V' = \left\{ w \in V \mid L \cap L_{x_0} \cap L_{x_1} \in D \right\}. \]

Because \( D \) blankets all of the \( L_{x_0} \)'s, we conclude that \( V' \) is a vertex cover of \( G \) of cardinality \( |V'| = |D| \leq K \), as desired.

The theorem follows.

4. Cover Relations

In this section, we examine properties of cover relations in linear orders and their consequences for poset covers.

Let \( P = (X, \prec P) \) be a poset, thought of as a transitive DAG. Then, a topological sort of \( P \) yields an order \( x_1, x_2, \ldots, x_n \) on \( X \) such that \( x_i \prec P x_j \) implies \( i < j \). Assume that \( P \) is not a linear order. Then there exist \( a, b \in X \) such that \( a \prec P b \). There exists at least one topological sort of \( P \) in which \( a \) appears to the left of \( b \), and there exists at least one topological sort of \( P \) in which \( b \) appears to the left of \( a \). This follows from alternate choices available to the depth-first search used to construct a topological sort. See [41]. Select a topological sort that makes \( a = x_i \) and \( b = x_j \), where \( i < j \). In that case, we obtain a proper extension \( P' \) of \( P \) in which \( a = x_i \), \( x_j = b \) by adding \( (x_i, x_j) \) to the DAG and taking the transitive closure. Moreover, we have \( a \prec_P b \), since the existence of \( c \) such that \( a \prec_P c \prec_P b \) contradicts \( a \prec_P b \). We have just demonstrated the following.

**Lemma 5** Let \( P = (X, \prec P) \) be a poset, and let \( a, b \in X \) satisfy \( a \prec_P b \). Let

\[ \prec_P = \prec P \cup \{(a, b)\} \cup \{(x, y) \mid x \prec_P a \text{ and } b \prec_P y\}. \]

Then \( P' = (X, \prec P') \) is a poset, \( P \subseteq P' \), and \( a \prec_P b \).

**Theorem 6** Let \( P = (X, \prec P) \) be a poset that is not a linear order, and let \( a, b \in X \) satisfy \( a \prec_P b \). Then there exists a proper extension \( P' = (X, \prec P') \) of \( P \) such that \( a \prec_P b \).

**Proof.** First, suppose that there exists a \( c \in X \) that is incomparable to \( a \). By Lemma 5, there exists a poset \( P' \) such that \( P \subseteq P' \) and \( c \prec_P a \). Moreover, \( c \prec_P a \prec_P b \), so by the definition of \( \prec_P, a \prec_P b \).

Second, the case of there being a \( c \in X \) that is incomparable to \( b \) is handled analogously.

Finally, we have the case that no element is incomparable to \( a \) or \( b \). Let \( c, d \in X \) be such that \( \{a, b\} \cap \{c, d\} = \emptyset \) and \( c \not\prec_P d \). (Such a pair \( c, d \) must exist, since \( P \) is not a linear order.) If either \( c \prec_P a \) and \( d \prec_P a \) or \( b \prec_P c \) and \( b \prec_P d \), then adding \( (c, d) \) to the DAG for \( P \) and taking the transitive closure gives us the desired poset. The case \( c \prec_P a \) and \( b \prec_P d \) (or vice versa) is impossible, since \( c \not\prec_P d \) and \( a \prec_P b \). There are no other cases, since \( a \prec_P b \).

The theorem follows.

**Theorem 7** Let \( P = (X, \prec P) \) be a poset, and let \( a, b \in X \) satisfy \( a \prec P b \). Then there exists a linear order \( L = (X, \prec_L) \) such that \( L \in \mathcal{L}(P) \) and \( a \prec_L b \). Moreover, for every linear extension \( L_1 = (X, \prec_{L_1}) \) of \( P \) in which \( a \prec_{L_1} b \), there exists a unique linear extension \( L_2 = (X, \prec_{L_2}) \) of \( P \) such that \( L_1 \leftrightarrow L_2 \) and \( b \prec_{L_2} a \).

**Proof.** By Lemma 5, there exists a poset \( P' \) such that \( P \subseteq P' \) and \( a \prec_P b \). By applying Theorem 6 iteratively to \( P' \), we ultimately obtain a linear order \( L \) that is an extension of \( P' \) (and hence of \( P \)) such that \( a \prec_L b \).

Now, let \( L_i = (X, \prec_{L_i}) \) be a linear extension of \( P \) in which \( a \prec_{L_i} b \). Let \( i = \rho(a) \); then \( \rho(b) = i + 1 \). Let \( L_i = \text{Swap}[L_{i+1};i] = (X, \prec_{L_{i+1}}) \). Let

\[ L = L_1 \setminus \{(a, b)\} = (X, \prec_{L_2}). \]

Then \( P' \) is a poset on \( X \) such that \( P \subseteq P' \) and such that \( a \) and \( b \) are incomparable in \( P' \). Moreover, \( P' = L_1 \setminus \{(a, b)\} \), so \( L_2 \) is a linear extension of \( P \) in which \( b \prec_{L_2} a \).

The theorem follows.

**Theorem 8** Let \( Y \) be a set of linear orders on \( X \). Let \( L_1 = x_1, x_2, \ldots, x_n \) be an element of \( Y \). Let \( i \) satisfy \( 1 \leq i \leq n - 1 \). Let \( P = (X, \prec_P) \) be a partial order of \( Y \) including \( L_1 \). If \( \text{Swap}[L_{i+1};i] \notin Y \), then \( x_i \) and \( x_{i+1} \) are comparable in \( P \) and \( x_{i+1} \prec_P x_i \).

**Proof.** Suppose that \( \text{Swap}[L_{i+1};i] \notin Y \). First assume that \( x_i \) and \( x_{i+1} \) are incomparable in \( P \). Then it must be true that \( x_i \prec_P x_{i+1} \), since \( x_{i+1} \) covers \( x_i \) in \( L \). For the same reason, there is no \( f \in \{1, 2, \ldots, i-1, i+2, \ldots, n\} \) such that \( x_i \prec_P x_f \). Hence, \( x_i \prec_P x_{i+1} \).

It remains to show that \( x_i \) and \( x_{i+1} \) are comparable in \( P \). To obtain a contradiction, assume that \( x_i \) and \( x_{i+1} \) are incomparable in \( P \). By Theorem 7, there exists a unique linear extension \( L' = (X, \prec_{L'}) \) of \( P \) such that \( L' \not\subseteq L \) and \( x_i \prec_{L'} x_{i+1} \). Necessarily, \( L' = \text{Swap}[L_{i+1};i] \).

Since \( L' \in \mathcal{L}(P) \) but \( L' \not\in Y \), we have a contradiction to the fact that \( P = (X, \prec_P) \) is a partial cover of \( Y \) including \( L \). The contradiction establishes that \( x_i \) and \( x_{i+1} \) are comparable in \( P \). The theorem follows.

We next characterize a set \( Y \) of linear orders that is covered by a single poset. The ordered pair \( (a, b) \) is a cover relation for \( Y \) if there exists an \( L \in Y \) and an \( L' \not\in Y \) such that \( \text{SwapPair}(L, L') = \{a, b\} \), \( a \prec_P b \) and \( b \prec_P a \). If \( (a, b) \) is a cover relation for \( Y \), then any poset \( P \) that partially covers \( Y \) including \( L \) must satisfy \( a \prec_P b \). An \( (a, b) \) cover sequence of length \( k \geq 2 \) for \( Y \) is a sequence \( a = c_1, c_2, \ldots, c_k = b \) such that \( (c_i, c_{i+1}) \) is a cover relation for \( Y \), for \( 1 \leq i \leq k - 1 \). If there is an \( (a, b) \) cover sequence for \( Y \), then any poset \( P \) that covers \( Y \) must satisfy \( a \prec_P b \).

**Theorem 9** A set \( Y \) of linear orders is the set of linear extensions of a single poset if and only if, for every \( a, b \in X \) for which \( a \neq b \), exactly one of the following holds: 1) \( \{a, b\} = \text{SwapPair}(L, L') \) for some \( L, L' \in Y \);
2) there is an \((a,b)\) cover sequence for \(Y\); or 3) there is a \((b,a)\) cover sequence for \(Y\).

**Proof.** For one direction, assume that there exists a poset \(P\) such that \(Y\) is the set of linear extensions of \(P\). Let \(a,b \in X\) satisfy \(a \neq b\).

First, suppose \(a \parallel b\). By Theorem 7, there exists a linear extension \(L\) of \(P\) for which \(a <_L b\) and another linear extension \(L'\) of \(P\) for which \(b <_L a\) and \(L \leftrightarrow L'\). Then, 1) holds. Neither 2) nor 3) holds, since those imply that \(a \neq b\) and \(b \neq a\) are comparable in \(P\).

Now suppose that \(a <_P b\). (The case \(b <_P a\) is symmetric.) Then 1) does not hold, since that implies that \(a \parallel b\). Also 3) does not hold, since that implies that \(b <_P a\). To demonstrate 2), it remains to construct an \((a,b)\) cover sequence for \(Y\). The first case is \(a <_P b\). Then, by repeated application of Theorem 6, there exists a linear extension \(L\) of \(P\) such that \(a <_L b\). Let \(L' = \text{Swap}(L; \{a,b\})\). Then, \(L' \not\subseteq Y\). Hence, \(a,b\) is an \((a,b)\) cover sequence for \(Y\). More generally, we can write \(a = c_1 < c_2 < \cdots < c_k = b\) for some \(c_1, c_2, \ldots, c_k\) such that \(c_i <_P c_{i+1}\) for \(1 \leq i < k - 1\). Then \(a = c_1, c_2, \ldots, c_k = b\) is also an \((a,b)\) cover sequence for \(Y\).

For the other direction, assume that, for every \(a,b \in X\) for which \(a \neq b\), exactly one of 1), 2), or 3) holds. Take \(P\) to be the poset generated by all the ordered pairs \((a,b)\) such that \(a \neq b\) is a cover sequence for \(Y\). We need to show that \(Y\) equals the set of linear extensions of \(P\). There are two cases to consider for each linear order \(L\). Let \(L = x_1, x_2, \ldots, x_n\).

**Case 1.**

\(L \in Y\). To obtain a contradiction, assume that \(L\) is not a linear extension of \(P\). Then there exist \(x_i\) and \(x_{i+1}\) such that \(x_{i+1} <_L x_i\). Let \(x_{i+1} = c_1 < c_2 < \cdots < c_k = x_i\) satisfy \(c_i <_P c_{i+1}\) for \(1 \leq i < k - 1\). Then \(x_{i+1} = c_1 < c_2 < \cdots < c_k = x_i\) is an \((x_{i+1}, x_i)\) cover sequence for \(Y\) and hence 2) holds for \(x_{i+1}\) and \(x_i\), but not 1) or 3). Let \(L' = \text{Swap}(L; i)\). Since 1) does not hold, we have \(L' \not\subseteq Y\). But then \(x_i, x_{i+1}\) is a cover sequence for \(Y\), a contradiction to the fact that 3) does not hold. In this case, we conclude that \(L\) is a linear extension of \(P\).

**Case 2.**

\(L \not\in Y\). Without loss of generality, we may assume that there exist \(L'\) and \(i\) such that \(L' \not\subseteq Y\) and \(L = \text{Swap}(L; i)\). Since \(x_{i+1} <_L x_i\), we have that \((x_{i+1}, x_i)\) is a cover sequence for \(Y\). Hence, \(x_{i+1} <_L x_i\) and \(L\) cannot be a linear extension of \(P\).

We conclude that \(Y\) is precisely the set of linear extensions of \(P\).

The theorem follows.

### 5. A Partial Cover Algorithm

In this section, we present an algorithm for finding a poset that is a partial cover with a maximal set of linear extensions.

#### 5.1. Some Intuition

Intuition for designing an algorithm to find a partial poset cover for a set \(Y\) of linear orders is developed first. It suffices to take a single \(L \in Y\) and identify a single poset \(P\) that is a partial cover of \(Y\) including \(L\). Observe that \(L\) is such a poset but is not satisfactory if we can construct a poset \(P \neq L\) that covers more of \(Y\). We use the swap graph \(G(Y)\) to direct construction of a more satisfactory \(P\).

During the process of constructing \(P\), we maintain a specification for a set of posets, each of which covers a constructed set \(Y' \subseteq Y\). We also maintain a set \(\Lambda \subseteq Y'\) consisting of linear orders, including \(L\), that have already been chosen to be covered by the final constructed poset. This specification consists of two kinds of information: some \(\prec\) relations and some \(\parallel\) relations. These relations must be consistent, that is, there must be at least one poset that satisfies them all. A bit more formally, the \(\prec\) relations can be specified by a set \(A \subseteq X \times X\) of ordered pairs, while the \(\parallel\) relations can be specified by a set \(B \subseteq \{a, b\}\) of unordered pairs. The specified set of posets is then \((Up(A)) \cap (Down(B))\).

\(A\) will be maintained to satisfy the following property. Let \(L \in \Lambda\) be arbitrary, and let \(a \prec b\) be any cover relation of \(L\). Let \(L' = \text{Swap}(L; \{a,b\})\). If \(L' \not\subseteq Y'\), then we require that \((a,b) \in A\). The rational for this requirement is that every poset \(P\) that covers \(L\) and does not cover \(L'\) satisfies the relation \(a \prec b\). As a side effect, every \(L' \in Y'\) for which \(b \prec a\) can be eliminated from further consideration for inclusion in \(\Lambda\).

\(B\) will be maintained to satisfy the following property. Again, let \(L \in \Lambda\) be arbitrary, and let \(a \parallel b\) be any cover relation of \(L\). Let \(L' = \text{Swap}(L; \{a,b\})\). If \(L' \in \Lambda\), then we require that \((a,b) \not\in B\). The rational for this requirement is that every poset \(P\) that covers both \(L\) and \(L'\) satisfies the relation \(a \parallel b\). As a side effect, every \(L' \in Y'\) for which the \((a,b)\) adjacency is not in \(G(Y')\) can be eliminated from further consideration for inclusion in \(\Lambda\).

We will need these definitions. Let \(a,b \in X\) be distinct, and let \(L\) be a linear order. The \((a,b)\)-interchange of \(L\) is the linear order that is the same as \(L\) except \(a\) and \(b\) have been exchanged. Let \(L_0, L_1, \ldots, L_k\) be a sequence of linear orders such that \(L_i \leftrightarrow L_{i+1}\), for \(0 \leq i < k - 1\), so that the sequence is a path \(Q\) in \(G(L(\emptyset))\). Let \(B\) be a subset of \(\{\{a,b\}\} \{a,b \in X \text{ and } a \neq b\}\). \(Q\) is \(B\)-labeled if, for \(0 \leq i < k - 1\), \(\text{SwapPair}(L_i, L_{i+1}) \in B\). A path
$Q' = L_{c_1}, L_{c_2}, \ldots, L_{c_k}$, in $G(L(\emptyset))$ is the \{a,b\} -mirror path of $Q$ if, for $0 \leq i \leq k$, $L_{c_i}$ is the \{a,b\} -interchange of $L_{c}$.

\section{5.2. The Algorithm}

\textbf{Figure 2} contains pseudocode for the algorithm Partial-Cover $(Y, L)$. It works by adding linear orders from $Y' \setminus \Lambda$ to $\Lambda$ one at a time, while maintaining the required properties for $A$ and $B$. The subroutine Trim in \textbf{Figure 3} is used to ensure that the required property for $A$ is maintained. The addition of a linear order to $\Lambda$ (Step 9) can add at most one new unordered pair to $B$ (Step 10).

We illustrate the algorithm with the example having $Y = \{12345, 21345, 32145, 31245, 13245, 12354, 21354, 23154, 13254\}$ and $L = 12345$. \textbf{Figure 4} contains the swap graph.

The call to Trim in Step 5 finds that $12435$ is not in $Y'$, so any linear orders in $\Lambda$ for which $4$ is less than $3$ should be deleted. In this case, there is no such linear order in $Y'$. After Step 6, $A = \{(3,4)\}$ and $B = \emptyset$.

The first time that Step 8 is executed in Partial-Cover, $A = \{(3,4)\}$ and $L = 12345$. (There are three choices for $L$. This is just one of them.) Then $\Lambda = \{12345, 21345\}$ (Step 9) and $B = \{(1,2)\}$ (Step 10). The call to Trim in Step 11 finds that $21345$ is not in $Y'$. The resulting cover relation $(3,4)$ is new, so $A$ is extended to $A = \{(3,4),(3,4)\}$. None of the linear orders in $Y'$ has $4$ less than $3$, so the call to Trim does not change $Y'$. The while loop from Steps 13 to 21 now has the swap pair $\{1,2\}$ to work with. Linear order $32154$ is missing its $\{1,2\}$ swap partner, $13254$. Hence, $32154$ is deleted from $Y'$, which is now

$Y' = \{12345, 21345, 23145, 32145, 31245, 13245, 12354, 21354, 23154, 13254\}$.

The second time that Step 8 is executed, $L_1 = 21345$ and $L_2 = 23145$. Then $\Lambda = \{12345, 21345, 23145\}$ (Step 9) and $B = \{(1,2),(1,3)\}$ (Step 10). The call to Trim in Step 11 finds that $32145$ is not in $Y'$. The resulting cover relation $(3,4)$ is new, so $A$ is extended to $A = \{(1,4),(3,4),(3,4)\}$. None of the linear orders in $Y'$ has $5$ less than $3$, so the call to Trim does not change $Y'$. The while loop from Steps 13 to 21 now has the swap pair $\{1,3\}$ to work with. Linear order $13254$ is missing its $\{1,3\}$ swap partner, $31254$. Hence, $13254$ is deleted from $Y'$, which is now

$Y' = \{12345, 21345, 23145, 32145, 31245, 13245, 12354, 21354, 23154, 13254\}$.

The third time that Step 8 is executed, $L_1 = 21345$ and $L_2 = 21354$. Then $\Lambda = \{12345, 21345, 23145, 21534\}$ (Step 9) and $B = \{(1,2),(1,3),(4,5)\}$ (Step 10). The call to Trim in Step 11 finds that $32145$ is not in $Y'$. The resulting cover relation $(3,5)$ is new, so $A$ is extended to $A = \{(1,4),(3,4),(3,5)\}$. None of the linear orders in $Y'$ has $5$ less than $3$, so the call to Trim does not change $Y'$. The while loop from Steps 13 to 21 now has the swap pair $\{4,5\}$ to work with. Linear orders $32145$, $31245$, and $13245$ are missing their $\{4,5\}$ swap part-

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure2}
\caption{Pseudocode for partial-cover $(Y, L)$.}
\end{figure}
Figure 3. Pseudocode for trim \((A, Y', \Lambda)\).

\[
\text{TRIM}(A, Y', \Lambda)
\]
1. Ensure that \(Y' \in \mathcal{L}(\min \Lambda)\)
2. done \(-\) FALSE
3. while NOT done
4. done \(-\) TRUE
5. do
6. for \(L \in \Lambda\)
7. do \(L' \leftarrow \text{Swap}[L; i]\)
8. if \(L' \not\in Y'\)
9. then \(A \leftarrow A \cup \{(L[i], L[i + 1])\}\)
10. for \(L \in Y'\)
11. do if \(L \not\in \min(A)\)
12. then \(Y' \leftarrow Y' \setminus \{L\}\)
13. done \(-\) FALSE
14. return \((A, Y')\)

Figure 4. Swap graph for example.

The fourth time that Step 8 is executed, \(L_1 = 21354\) and \(L_2 = 21354\). Then \(\Lambda = \{12345, 21345, 23145, 12354, 21354, 23154\}\) (Step 9) and \(B = \{1, 2\}, \{1, 3\}, \{4, 5\}\) (Step 10). The call to Trim in Step 11 finds that 31254 and 23514 are not in \(Y'\). The resulting cover relations are \((2, 3)\), which is not new, and \((1, 5)\), which is new, so \(A\) is extended to \(A = \{(1, 4), (1, 5), (2, 3), (3, 4), (3, 5)\}\). None of the linear orders in \(Y'\) has 3 less than 2 or 5 less than 1, so the call to Trim does not change \(Y'\). The while loop from Steps 13 to 21 has no new swap pairs to work with. Hence, there are no further linear orders to delete from \(Y'\), which remains

\(Y' = \{12345, 21345, 23145, 12354, 21354, 23154\}\).

The fifth and last time that Step 8 is executed, \(L_1 = 21354\) and \(L_2 = 21354\). Then \(\Lambda = \{12345, 21345, 23145, 12354, 21354, 23154\}\) (Step 9) and \(B = \{1, 2\}, \{1, 3\}, \{4, 5\}\) (Step 10). The call to Trim in Step 11 finds that 31254 and 23514 are not in \(Y'\). The resulting cover relations are \((2, 3)\), which is not new, and \((1, 5)\), which is new, so \(A\) is extended to \(A = \{(1, 4), (1, 5), (2, 3), (3, 4), (3, 5)\}\). None of the linear orders in \(Y'\) has 3 less than 2 or 5 less than 1, so the call to Trim does not change \(Y'\). The while loop from Steps 13 to 21 has no new swap pairs to work with. Hence, there are no further linear orders to delete from \(Y'\), which remains

\(Y' = \{12345, 21345, 23145, 12354, 21354, 23154\}\).

At this point, \(\Lambda = Y'\).

The resulting poset \(P\) has the cover relations in \(A\), namely, \((1, 4), (1, 5), (2, 3), (3, 4), (3, 5)\), and \((1, 5)\). The set of linear extensions of \(P\) is exactly the final value of \(Y'\), namely, \(\{12345, 21345, 23145, 12354, 21354, 23154\}\).

5.3. Proof of Correctness

We assume that the following loop invariants hold each time that the test at the top of the while loop body (Step 7) is executed.

1. \(L \in Y \subseteq Y' \subseteq \mathcal{Y}\).
2. \(\mathcal{G}(Y')\) is a connected graph, and \(\mathcal{G}(\Lambda)\) is a connected graph.
3. The directed graph \((X, A)\) contains no cycles.
4. Every element of \(Y'\) is a linear extension of \(\min(A)\), that is, \(Y' \subseteq \mathcal{L}(\min(A))\).
5. The set \(A\) equals the set of ordered pairs \((a, b) \in X \times X\) such that there exist \(L' \in \Lambda\) such that \(\text{Swap}[L'; (a, b)] \not\in Y'\).
6. \(\min(A) \in \text{Down}(B)\) and consequently \(\text{Up}(A) \cap \text{Down}(B) \neq \emptyset\).
7. The set \(B\) equals the set of unordered pairs \((a, b) \in X\) such that \(a \neq b\) and such that there exist \(L', L'' \in \Lambda\) satisfying \(\text{SwapPair}(L'; (a, b)) = (a, b)\).
8. Let \(Q = L_0, L_1, \ldots, L_k\) be a \(B\)-labeled path of linear orders such that \(L_0 \in Y', a \prec_{L_0} b,\) and \(a \prec_{L_0} b\) and such that \(Q\) is a shortest \(B\)-labeled path from \(L_0\) to \(L_k\). Let \(Q' = L_0, L_1, \ldots, L_k\) be the \([a, b]\)-mirror path for \(Q\). Then, all of the \(L_i\)'s are in \(Y'\), and either all of the \(L_i\)'s are in \(Y'\) or none of them are.

Together, these invariants suffice to demonstrate the correctness of Partial-Cover, culminating in Theorem 11.

Every execution of subroutine Trim enforces Invariant...
Invariant 2 is guaranteed by Steps 6 and 22 and by the way that linear orders are selected for addition to $\Lambda$ (Step 8).

Invariants 1 and 2 guarantee that, whenever Step 8 is reached, there is a suitable $L_i, L_j$ pair to select.

The fact that $Y' \subseteq Y$ is guaranteed through the initialization in Step 3 and the fact that any change to $Y'$ always selects a subset of $Y'$.

**Initialization.** After initialization (Steps 2 through 6), all invariants are true for the first execution of Step 7, for the following reasons. We have $\Lambda = \{L_i\}$ and $B = \emptyset$. The execution of Trim (Step 5) ensures that Invariants 4 and 5 hold, while maintaining $L_j \in \Lambda \subseteq Y'$ (Invariant 1). Step 6 guarantees Invariant 2. Invariant 3 holds because the only order relations in $A$ are cover relations in $L_j$.

The fact that $B = \emptyset$ makes Invariants 6, 7, and 8 true vacuously.

**Execution of the loop body.** The fact that $\Lambda \subseteq Y'$ requires that the algorithm never deletes an element of $\Lambda$ from $Y'$ in Step 19 or in Trim. That fact also implies $L_j \in \Lambda$, since $L_j$ is initially put in $\Lambda$ (Step 2) and could only be deleted in Step 19 or in Trim.

The algorithm never deletes an element of $\Lambda$ from $Y'$ in Trim. To obtain a contradiction, assume that $L_j \in \Lambda$ is deleted in Step 12 of Trim and that it is the first element of $\Lambda$ deleted. The deletion of $L_j$ is caused by a sequence $a = c_i, c_2, \ldots, c_k = b$ such that $k \geq 2, (c_i, c_{i+1}) \in A$, for $1 \leq i \leq k - 1$, and $b <_{L_j} a$. For each $i$ satisfying $1 \leq i \leq k - 1$, there exists an $\hat{L}_i \in \Lambda$ such that $c_i <_{\hat{L}_i} c_{i+1}$, and $\text{Swap}[\hat{L}_i; (c_i, c_{i+1})] \notin Y'$. There is a path in $G(\Lambda)$ from $L_j$ to $\hat{L}_i$ that does not contain an edge with swap pair $(c_i, c_{i+1})$, since $(c_i, c_{i+1}) \in B$ contradicts $(c_i, c_{i+1}) \in A$. Consequently, $c_i$ and $c_{i+1}$ are in the same order in $L_j$ and in $\hat{L}_i$, which implies that $c_i <_{L_j} c_{i+1}$. Taken together, these relations imply that $a <_{L_j} b$, a contradiction to $b <_{L_j} a$.

We conclude that $L_j$ is, in fact, not deleted in Trim.

The algorithm never deletes an element of $\Lambda$ from $Y'$ in Step 19. The deletion of an element $L_j \in Y'$ depends on the swap pairs in $B$. More particularly, such a deletion would require a $B$-labeled path in $G(Y')$ labeled from $B$ to $L_j$ to some $L_i$ that has a swap pair from $B$ that goes to a linear order outside $Y'$.

This cannot happen because of Invariant 8. We conclude that $L_j$ is, in fact, not deleted in Step 19.

Invariant 3 is maintained because the existence of a cycle in $A$ implies that $A$ and $B$ are inconsistent.

Invariants 4 and 5 are maintained by Trim.

The consistency of $A$ and $B$ (Invariant 6) is maintained by Trim and the loop at Steps 15 through 21.

Invariant 7 is maintained by the way that elements are added to $B$ (Step 10).

It remains to show that Invariant 8 holds; this is demonstrated in the following lemma.

**Lemma 10** Each time that Step 8 is about to be executed, Invariant 8 holds.

Proof. Let $P = L_0, L_1, \ldots, L_k$ be a $B$-labeled path of linear orders such that $L_0 \in Y'$, $a <_{L_0} b$, and $a <_{L_0} b$ and such that $P$ is a shortest $B$-labeled path from $L_0$ to $L_k$. Let $P' = L_0, L_1, \ldots, L_k'$ be the $\{a, b\}$-mirror path for $P$.

To obtain a contradiction, assume that there is some $L_j$ that is not in $Y$. Let $L_j$ be the first such. Then $i \neq 0$, so $L_{i-1} \in Y$. Let $\{c, d\} = \text{SwapPair}(L_{i-1}, L_i) \in B$. Since $L_i \notin Y$, $L_{i-1}$ cannot be in $Y$, since it would have been deleted in a previous iteration due to the swap pair $\{c, d\}$ being in $B$. This is a contradiction. We conclude that all of the $L_j$’s are in $Y$.

We next show that $P'$ is not just a shortest $B$-labeled path but is also a shortest path in $G(L(\mathcal{O}))$. Let $C = \{(c, d)\} | c <_{L_0} d$ and $d <_{L_0} c\}$. For any path from $L_0$ to $L_k$ in $G(L(\mathcal{O}))$, every $\{c, d\} \in C$ must be the swap pair for some edge in the path. Consequently, $C \subseteq B$. Moreover, there is a path in $G(L(\mathcal{O}))$ that uses swap pairs only from $C$ and each only once, so the length of every shortest path from $L_0$ to $L_k$ is $|C|$. (Think about the swaps done by bubble sort; these give one such shortest path.) Since $P$ is a shortest $B$-labeled path from $L_0$ to $L_k$, it must be a $C$-labeled path having $k = |C|$. Note that, therefore, no swap pair occurs more than once in $P$.

We next show that $P'$ is a $B$-labeled path. Since $P$ contains no swap pair more than once and since $a <_{L_0} b$ and $a <_{L_0} b$, if there is a swap pair $\{a, x\}$ labeling an edge of $P'$, we must also have the swap pair $\{b, x\}$ labeling another edge of $P$, and vice versa. More succinctly, $\{a, x\} \in C$ if and only if $\{b, x\} \in C$.

Let $\{c, d\} = \text{SwapPair}(L_i, L_{i+1})$, for some $i$ satisfying $0 \leq i \leq k - 1$. If $\{c, d\} \cap \{a, b\} = \emptyset$, then $\{c, d\} = \text{SwapPair}(L_i, L_{i+1})$. If $\{c, d\} = \{a, d\}$, then $\{b, d\} = \text{SwapPair}(L_i, L_{i+1})$, which is in $C$ by the argument above. Similarly, if $\{c, d\} = \{b, d\}$, then $\{a, d\} = \text{SwapPair}(L_i, L_{i+1})$, which is in $C$ by the argument above. We conclude that $P'$ is a $C$-labeled path and hence a $B$-labeled path.

Finally, we show that either all of the $L_j$’s are in $Y'$ or none of them are. To obtain a contradiction, suppose that $L_j \in Y'$ and that $L_{i-1} \notin Y'$, for some $i$ satisfying $0 \leq i \leq k - 1$. (The case $L_j \notin Y'$ and $L_{i-1} \in Y'$ will yield a contradiction by an analogous argument.) But, in this case, $L_j$ would have been deleted from $Y'$ in an earlier iteration, a contradiction. From this contradiction, we conclude that either all of the $L_j$’s are in $Y'$ or none of them are.
Hence, Invariant 8 holds.

The correctness and time complexity of the algorithm are now in view.

**Theorem 11** Algorithm Partial-Cover ($\mathcal{Y}, L$) returns a set $A$ such that $\text{Min}(A)$ is a partial cover of $\mathcal{Y}$ including $L$. The algorithm has time complexity $O(n^2 |\mathcal{Y}|^2)$.

**Proof.** The correctness of the algorithm follows from the prior discussion of the loop invariants.

For the time complexity, we first note that $\lvert A \rvert = O(n^2)$ and $\lvert B \rvert = O(n^2)$.

We examine the subroutine Trim. Trim is executed once in Step 5; once for each addition to $\Lambda$ (Step 11; $O(|\mathcal{Y}|)$ times in total); and once for each deletion in Step 19 (Step 20; $O(|\mathcal{Y}|)$ times in total). Hence, Trim is executed $O(|\mathcal{Y}|)$ times. The loop in Steps 5 through 9 requires $O(n^2 |\mathcal{Y}|)$ time for one execution. The time complexity to test coverage in Step 11 requires $O(|A|) = O(n^2)$ time. Hence, the loop in Steps 10 through 13 requires $O(n^2 |\mathcal{Y}|)$ time for one execution. The while loop is executed $O(|\mathcal{Y}|)$ times, since each additional iteration of the loop because done $=$ False requires the reduction of the cardinality of $\mathcal{Y}$ by at least one. We conclude that the total time spent in trim is $O(n^2 |\mathcal{Y}|^2)$.

It is easy to check that the complexity bound for all calls to Trim dominates the time complexity of the algorithm. Hence, the time complexity of Partial-Cover is $O(n^2 |\mathcal{Y}|^2)$, as required.

**REFERENCES**


