Open Journal of Discrete Mathematics, **2013, 3, 101-111**
http://dx.doi.org/10.4236/ojdm.2013.33020 Published Online July 2013 (http://www.scirp.org/journal/ojdm)

The Poset Cover Problem

Lenwood S. Heath[1], Ajit Kumar Nema[2]
[1]Department of Computer Science, Virginia Tech, Blacksburg, USA
[2]Core 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 $P=(X,<)$ 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 $\{P_1, P_2, \cdots, P_k\}$ of posets on X covers the set of linear orders that is the union of the $\mathcal{L}(P_i)$. Given linear orders $L_1, L_2, \cdots, L_m$ on X, the Poset Cover problem is to determine the smallest number of posets that cover $\{L_1, L_2, \cdots, L_m\}$. 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, antisymmetric, and transitive binary relation $<_P$ defined on a finite set X of cardinality $n \geq 1$. We write P as the ordered pair $P=(X,<_P)$. Equivalently, poset P is a transitive directed acyclic graph (DAG), namely, $P=(X,\{(x,y)|x<_P y\})$. If G is a DAG, then its transitive closure is a poset by this equivalence. The *rank function* $\rho_P: X\to\{1,2,\cdots,n\}$ is given by $\rho_P(x)=1+|\{y|y<_P x\}|$. The *empty poset* is $\epsilon=(X,\varnothing)$.

Let $x,y\in X$ be distinct. Then x and y are *comparable in* P, written $x\perp_P y$, if $x<_P y$ or $y<_P x$, while x and y are *incomparable*, written $x\|_P y$, otherwise. Moreover, x is *covered* by y or y *covers* x, written $x\prec_P y$, if $x<_P y$ and there is no $z\in X$ such that $x<_P z<_P y$. 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 $P_1=(X,<_{P_1})$ and $P_2=(X,<_{P_2})$ are posets on the same set X, then P_2 is an *extension* of P_1, written $P_1\sqsubseteq P_2$, if, for all $a,b\in X, a<_{P_1} b$ implies $a<_{P_2} b$. The relation $\sqsubseteq$ on posets of X is reflexive, antisymmetric, and transitive.

A *linear order* $L=(X,<_L)$ on X is a poset L such that, for $x,y\in X$, either $x=y$ or $x\perp_L y$ holds. If L is a linear order, then the rank function $\rho_L: X\to\{1,2,\cdots,n\}$ is a bijection. Setting $x_i=\rho_L^{-1}(i)$, L can be written as the sequence $L=x_1,x_2,\cdots,x_n$, 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 $P\sqsubseteq L$. 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 $\mathcal{P}=\{P_1,P_2,\cdots,P_k\}$ be a set of k posets on X. This set *covers* the set of linear orders

$$\Upsilon=\bigcup_{i=1}^{k}\mathcal{L}(P_i).$$

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

$$\Upsilon\subseteq\bigcup_{P\in\mathcal{P}}\mathcal{L}(P).$$

Lemma 1 *Let Υ 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 $k'\le k$ that also covers Υ such that every poset in $\hat{\mathcal{P}}$ is maximal in Υ.*

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

The constructed $\hat{\mathcal{P}}$ has cardinality $\le k$. Moreover, $\hat{\mathcal{P}}$ also covers Υ and every poset in $\hat{\mathcal{P}}$ is maximal in Υ. 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 $n\ge 1$; a nonempty set $\Upsilon=\{L_1,L_2,\cdots,L_m\}$ of linear orders over X; and an integer $K\le m$.

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

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

Let $L=x_1,x_2,\cdots,x_n$ be a linear order on X. For each i satisfying $1\le i\le n-1$, the *i-swap* of L is the linear order $\text{Swap}[L;i]=x_1,x_2,\cdots,x_{i-1},x_{i+1},x_i,x_{i+2},\cdots,x_n$. Let $L'=\text{Swap}[L;i]$. Evidently, $L=\text{Swap}[L';i]$, so the i-swap relation is symmetric, written $L\overset{i}{\longleftrightarrow}L'$. For pairs (L,L') that are i-swaps of each other, for some i, we define the function $\text{SwapIndex}(L,L')=i$. Note that the set differences of L and L', namely $L\setminus L'=\{(x_i,x_{i+1})\}$ and $L'\setminus L=\{(x_{i+1},x_i)\}$, 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, $\text{SwapPair}(L,L')=\varnothing$. Two linear orders L_1 and L_2 *differ by a swap*, written $L_1\leftrightarrow L_2$, if $L_1\overset{i}{\longleftrightarrow}L_2$, for some i. Since $L_1\leftrightarrow L_2$ if and only if $L_1\leftrightarrow L_2$, the $\leftrightarrow$ relation is also symmetric. If $L_1\leftrightarrow L_2$, $a\prec_{L_1} b$, and $b\prec_{L_2} a$, then we write $L_2=\text{Swap}[L_1;\{a,b\}]$ to mean that $L_1\overset{i}{\longleftrightarrow}L_2$ for some i, where the elements swapped are a and b.

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

Let Π 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

$$\mathrm{Up}(A) = \{P \in \Pi \mid a <_P b \text{ for all } (a,b) \in A\}.$$

$\mathrm{Up}(A)$ is empty if and only if the directed graph (X, A) contains cycles. Let $B \subseteq \{\{a,b\} \mid a,b \in X \text{ and } a \neq b\}$ be a set of unordered pairs. The *down-set* of B is

$$\mathrm{Down}(B) = \{P \in \Pi \mid a \|_P b \text{ for all } \{a,b\} \in B\}.$$

$\mathrm{Down}(\varnothing) = \Pi$, and we always have the empty poset $\epsilon \in \mathrm{Down}(B)$.

If $\mathrm{Up}(A) \neq \varnothing$, then define the *minimal element* in $\mathrm{Up}(A)$ to be

$$\mathrm{Min}(A) = \bigcap_{P \in \mathrm{Up}(A)} P.$$

The following properties of $\mathrm{Min}(A)$ follow directly from the definitions.

Lemma 2 *$A \subseteq \mathrm{Min}(A)$ and $\mathrm{Up}(A) = \{P \mid \mathrm{Min}(A) \sqsubseteq P\}$.*

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 $\mathrm{Up}(B) \subseteq \mathrm{Up}(A)$. Let $C, D \subseteq \{\{c,d\} \mid c,d \in X \text{ and } c \neq d\}$. If $C \subseteq D$, then $\mathrm{Down}(D) \subseteq \mathrm{Down}(C)$.*

Proof. Suppose that $A \subseteq B$. By the definition of up-sets,

$$\mathrm{Up}(B) = \{P \in \Pi \mid a <_P b \text{ for all } (a,b) \in B\}$$
$$\subseteq \{P \in \Pi \mid a <_P b \text{ for all } (a,b) \in A\} = \mathrm{Up}(A).$$

Now, suppose that $C \subseteq D$. Then,

$$\mathrm{Down}(D) = \{P \in \Pi \mid a \|_P b \text{ for all } \{a,b\} \in D\}$$
$$\subseteq \{P \in \Pi \mid a \|_P b \text{ for all } \{a,b\} \in C\} = \mathrm{Down}(C),$$

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 $V' \subset V$ 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, $\Upsilon = \{L_1, L_2, \cdots, L_m\}$, and K constitute an instance of Poset Cover, and let $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ 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 $\mathcal{P}$ covers Υ 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 $L \in \mathcal{L}(P_i)$ is generated, check whether $L \in \Upsilon$. If not, then return No. If so, then mark that element of Υ 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 Υ is marked Covered. If so, then return Yes; otherwise, return No. We find that the worst-case time to check whether $\mathcal{P}$ covers Υ 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 $V = \{1, 2, \cdots, \ell\}$. Let $s = |E| = 3\ell/2$, and let $E = \{e_1, e_2, \cdots, e_s\}$ 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 $X = \{x_1, x_2, \cdots, x_n\}$ be a base set of n elements. Let L_b, the *base order*, be the linear order on X specified by

$$x_1 <_{L_b} x_2 <_{L_b} \cdots <_{L_b} x_n.$$

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 Υ are obtained by a small set of swaps of such pairs, applied to L_b.

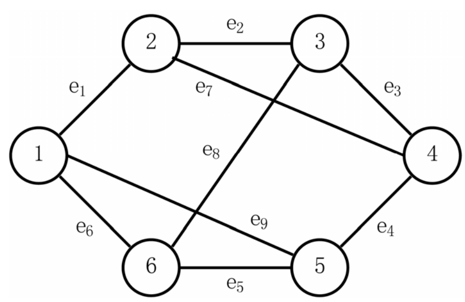

Figure 1. A cubic graph as part of an instance of cubic vertex cover.

The first s pairs correspond to the s edges in a natural way. In particular, edge $e_i \in E$ is associated with the *edge order* $L_{e_i} = \text{Swap}\left[L_b; 2i-1\right]$. Continuing the example, we set $n = 2(s+2) = 22$,

$$X = \{x_1, x_2, \cdots, x_{22}\}, L_b = x_1, x_2, \cdots, x_{22},$$

and, for example,

$$L_{e_1} = x_2, x_1, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}.$$

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

$$L_{e_{\chi[v,i]}, e_{\chi[v,j]}} = \text{Swap}\Big[\text{Swap}\big[L_b; 2\chi[v,i]-1\big]; 2\chi[v,j]-1\Big].$$

For the running example, there are 18 pair edge orders. For each triple $e_{\chi[v,1]}, e_{\chi[v,2]}$, and $e_{\chi[v,3]}$, we define the *triple edge order* to be

$$L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}} = \text{Swap}\Big[\text{Swap}\Big[\text{Swap}\big[L_b; 2\chi[v,i]-1\big]; 2\chi[v,j]-1\Big]; 2\chi[v,k]-1\Big]$$

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_i,e_j}, there is a corresponding *secondary pair edge order* obtained by swapping x_{2s+1} and x_{2s+2}, which is

$$L'_{e_i,e_j} = \text{Swap}\left[L_{e_i,e_j}; 2s+1\right].$$

For primary triple edge order L_{e_i,e_j,e_k}, there is a corresponding *secondary triple edge order* obtained by swapping x_{2s+3} and x_{2s+4}, which is

$$L'_{e_i,e_j,e_k} = \text{Swap}\left[L_{e_i,e_j,e_k}; 2s+3\right].$$

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:

$$A = \left\{L_{e_i} \middle| 1 \le i \le s\right\},$$

$$B = \left\{L_{e_i,e_j} \middle| e_i \text{ and } e_j \text{ are incident on some } v \in V\right\},$$

$$B' = \left\{L'_{e_i,e_j} \middle| e_i \text{ and } e_j \text{ are incident on some } v \in V\right\},$$

$$C = \left\{L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}} \middle| v \in V\right\},$$

$$C' = \left\{L'_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}} \middle| v \in V\right\}.$$

We can now complete our instance of Poset Cover by setting

$$\Upsilon = \{L_b\} \cup A \cup B \cup B' \cup C \cup C'$$

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

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, Υ, 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 Υ is maximal in Υ. Since the elements of B' must be blanketed by any cover, we may assume that the set

$$\mathcal{B}' = \left\{L_{e_i,e_j} \cap L'_{e_i,e_j} \middle| e_i \text{ and } e_j \text{ incident on some } v \in V\right\}$$

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

$$\mathcal{C}' = \left\{L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}} \cap L'_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}} \middle| v \in V\right\}$$

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

First, assume that $V' \subset V$ is a vertex cover of G of cardinality at most K. Define

$$\mathcal{P} = \mathcal{B}' \cup \mathcal{C}' \cup \bigcup_{v \in V'} \left\{L_b \cap L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}}\right\}.$$

Note that $|\mathcal{P}| = 4\ell + |V'| \le 4\ell + K = K'$ and that, by previous observations, it suffices to demonstrate that $\mathcal{P}$ blankets L_b and A. Since G is nonempty, $|E| > 0$, and $|V'| > 0$. Therefore, L_b is blanketed by each of the posets $L_b \cap L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}}$ in $\mathcal{P}$ corresponding to a vertex $v \in V'$. For an edge $e_i \in E$, there is a $v \in V'$ incident on e_i. Then, L_{e_i} is blanketed by the poset $L_b \cap L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}}$ in $\mathcal{P}$, and hence every linear order in A is blanketed by $\mathcal{P}$. We conclude that $\mathcal{P}$ covers Υ, as desired.

Now, assume that $\mathcal{P}$ is a cover of Υ of cardinality at most K'. By previous observations, we must have $\mathcal{B}' \cup \mathcal{C}' \cup \mathcal{D}'$, for some set $\mathcal{D}$ of cardinality at most K. Since $\mathcal{P}$ covers Υ, $\mathcal{D}$ must blanket L_b and A. Let e_i be any edge of G, incident on vertices u and v. Without loss of generality, we may assume that $i = \chi[u,1] = \chi[v,1]$. There are two maximal posets that blanket $L_{e_i} : L_b \cap L_{e_{\chi[u,1]}, e_{\chi[u,2]}, e_{\chi[u,3]}}$ and $L_b \cap L_{e_{\chi[v,1]}, e_{\chi[v,2]}, e_{\chi[v,3]}}$. 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_b, and the only other orders for $\mathcal{D}$ to blanket are the L_{e_i}'s.

Define

$$V' = \left\{ w \in V \,\middle|\, L_b \cap L_{e_{\chi[w,1]}, e_{\chi[w,2]}, e_{\chi[w,3]}} \in \mathcal{D} \right\}.$$

Because $\mathcal{D}$ blankets all of the L_{e_i}'s, we conclude that V' is a vertex cover of G of cardinality $|V'| = |\mathcal{D}| \le K$, 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 $P = (X, <_P)$ be a poset, thought of as a transitive DAG. Then, a topological sort of P yields an order $x_1, x_2, \cdots, x_n$ on X such that $x_i <_P x_j$ implies $i < j$. Assume that P is not a linear order. Then there exist $a, b \in X$ such that $a \|_P b$. 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 $a = x_i <_{P'} x_j = b$ 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 $a <_P c <_P b$ contradicts $a \|_P b$. We have just demonstrated the following.

Lemma 5 *Let* $P = (X, <_P)$ *be a poset, and let* $a, b \in X$ *satisfy* $a \|_P b$. *Let*

$$<_{P'} = <_P \cup \{(a,b)\} \cup \{(x,y) \mid x <_P a \text{ and } b <_P y\}.$$

Then $P' = (X, <_{P'})$ is a poset, $P \sqsubseteq P'$, and $a \prec_{P'} b$.

Theorem 6 *Let* $P = (X, <_P)$ *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* $P' = (X, <_{P'})$ *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 $P \sqsubseteq P'$ and $c \prec_{P'} a$. Moreover, $c <_{P'} a <_{P'} b$, so, by the definition of $<_{P'}, a \prec_{P'} b$.

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

The theorem follows.

Theorem 7 *Let* $P = (X, <_P)$ *be a poset, and let* $a, b \in X$ *satisfy* $a \|_P b$. *Then there exists a linear order* $L = (X, <_L)$ *such that* $L \in \mathcal{L}(P)$ *and* $a \prec_L b$. *Moreover, for every linear extension* $L_1 = (X, <_{L_1})$ *of* P *in which* $a \prec_{L_1} b$, *there exists a unique linear extension* $L_2 = (X, <_{L_2})$ *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 $P \sqsubseteq P'$ 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 $L_1 = (X, <_{L_1})$ be a linear extension of P in which $a \prec_{L_1} b$. Let $i = \rho_{L_1}(a)$; then $\rho_{L_1}(b) = i + 1$. Let $L_2 = \text{Swap}[L_1; i] = (X, <_{L_2})$. Let $P' = L_1 \setminus \{(a,b)\} = (X, <_{P'})$. Then P' is a poset on X such that $P \sqsubseteq P'$ and such that a and b are incomparable in P'. Moreover, $P' = L_2 \setminus \{(b,a)\}$, so L_2 is a linear extension of P in which $b \prec_2 a$.

The theorem follows.

Theorem 8 *Let* Υ *be a set of linear orders on* X. *Let* $L = x_1, x_2, \cdots, x_n$ *be an element of* Υ. *Let* i *satisfy* $1 \le i \le n-1$. *Let* $P = (X, <_P)$ *be a partial cover of* Υ *including* L. *If* $\text{Swap}[L; i] \notin \Upsilon$, *then* x_i *and* x_{i+1} *are comparable in* P *and* $x_i \prec_P x_{i+1}$.

Proof. Suppose that $\text{Swap}[L; i] \notin \Upsilon$. First assume that x_i and x_{i+1} are comparable in P. Then it must be true that $x_i <_P x_{i+1}$, since x_{i+1} covers x_i in L. For the same reason, there is no $j \in \{1, 2, \cdots, i-1, i+2, \cdots, n\}$ such that $x_i <_P x_j <_P x_{i+1}$. 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 $L' = (X, <_{L'})$ of P such that $L \leftrightarrow L'$ and $x_{i+1} \prec_{L'} x_i$. Necessarily, $L' = \text{Swap}[L; i]$. Since $L' \in \mathcal{L}(P)$ but $L' \notin \Upsilon$, we have a contradiction to the fact that $P = (X, <)$ is a partial cover of Υ including L. The contradiction establishes that x_i and x_{i+1} are comparable in P. The theorem follows.

We next characterize a set Υ of linear orders that is covered by a single poset. The ordered pair (a,b) is a *cover relation* for Υ if there exists an $L \in \Upsilon$ and an $L' \notin \Upsilon$ such that $\text{SwapPair}(L, L') = \{a, b\}$, $a \prec_L b$, and $b \prec_{L'} a$. If (a,b) is a cover relation for Υ, then any poset P that partially covers Υ including L must satisfy $a <_P b$. An (a,b) *cover sequence of length* $k \ge 2$ for Υ is a sequence $a = c_1, c_2, \cdots, c_k = b$ such that (c_i, c_{i+1}) is a cover relation for Υ, for $1 \le i \le k-1$. If there is an (a,b) cover sequence for Υ, then any poset P that covers Υ must satisfy $a <_P b$.

Theorem 9 *A set* Υ *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* $a \ne b$, *exactly one of the following holds*: 1) $\{a, b\} = \text{SwapPair}(L, L')$ *for some* $L, L' \in \Upsilon$;

2) *there is an* (a,b) *cover sequence for* Υ; *or* 3) *there is a* (b,a) *cover sequence for* Υ.

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

First, suppose $a \|_P b$. By Theorem 7, there exists a linear extension L of P for which $a \prec_L b$ and another linear extension L' of P for which $b \prec_L a$ and $L \leftrightarrow L'$. Then, 1) holds. Neither 2) nor 3) holds, since those imply that a and b 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 $a \|_P b$. 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 Υ. The first case is $a \prec_P b$. Then, by repeated application of Theorem 6, there exists a linear extension L of P such that $a \prec_P b$. Let $L' = \mathrm{Swap}[L;\{a,b\}]$. Then, $L' \notin \Upsilon$. Hence, a,b is an (a,b) cover sequence for Υ. More generally, we can write $a = c_1, c_2, \cdots, c_k = b$ for some $c_1, c_2, \cdots, c_k$ such that $c_i \prec_P c_{i+1}$ for $1 \le i \le k-1$. Then $a = c_1, c_2, \cdots, c_k = b$ is also an (a,b) cover sequence for Υ.

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,b is a cover sequence for Υ. We need to show that Υ equals the set of linear extensions of P. There are two cases to consider for each linear order L. Let $L = x_1, x_2, \cdots, x_n$.

Case 1.

$L \in \Upsilon$. 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 $x_{i+1} <_P x_i$. Let $x_{i+1} = c_1, c_2, \cdots, c_k = x_i$ satisfy $c_1 \prec_P c_2 \prec_P \cdots \prec_P c_k$. Then, $x_{i+1} = c_1, c_2, \cdots, c_k = x_i$ is an (x_{i+1}, x_i) cover sequence for Υ and hence 2) holds for x_{i+1} and x_i, but not 1) or 3). Let $L' = \mathrm{Swap}[L;i]$. Since 1) does not hold, we have $L' \notin \Upsilon$. But then x_i, x_{i+1} is a cover sequence for Υ, 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.

$L \notin \Upsilon$. Without loss of generality, we may assume that there exist L' and i such that $L' \in \Upsilon$ and $L = \mathrm{Swap}[L';i]$. Since $x_{i+1} <_{L'} x_i$, we have that (x_{i+1}, x_i) is a cover sequence for Υ. Hence, $x_{i+1} <_P x_i$ and L cannot be a linear extension of P.

We conclude that Υ 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 Υ of linear orders is developed first. It suffices to take a single $L \in \Upsilon$ and identify a single poset P that is a partial cover of Υ 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 Υ. We use the swap graph $\mathcal{G}(\Upsilon)$ 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 $\Upsilon' \subseteq \Upsilon$. We also maintain a set $\Lambda \subseteq \Upsilon'$ 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 $<$ relations and some $\|$ relations. These relations must be consistent, that is, there must be at least one poset that satisfies them all. A bit more formally, the $<$ relations can be specified by a set $A \subseteq X \times X$ of ordered pairs, while the $\|$ relations can be specified by a set $B \subseteq \{\{a,b\} \mid a,b \in X \text{ and } a \neq b\}$ of unordered pairs. The specified set of posets is then $\mathrm{Up}(A) \cap \mathrm{Down}(B)$.

A will be maintained to satisfy the following property. Let $L \in \Lambda$ be arbitrary, and let $a \prec_L b$ be any cover relation of L. Let $L' = \mathrm{Swap}[L;\{a,b\}]$. If $L' \notin \Upsilon'$, 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 $a <_P b$. As a side effect, every $L'' \in \Upsilon'$ for which $b <_{L''} a$ can be eliminated from further consideration for inclusion in Λ.

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

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 $L_0, L_1, \cdots, L_k$ be a sequence of linear orders such that $L_i \leftrightarrow L_{i+1}$, for $0 \le i \le k-1$, so that the sequence is a path Q in $\mathcal{G}(\mathcal{L}(\varnothing))$. Let B be a subset of $\{\{a,b\} \mid a,b \in X \text{ and } a \neq b\}$. Q is B*-labeled* if, for $0 \le i \le k-1, \mathrm{SwapPair}(L_i, L_{i+1}) \in B$. A path

$Q' = L_{0'}, L_{1'}, \cdots, L_{k'}$ in $\mathcal{G}(\mathcal{L}(\varnothing))$ is the $\{a,b\}$ *-mirror path* of Q if, for $0 \le i \le k$, $L_{i'}$ is the $\{a,b\}$ -interchange of L_i.

5.2. The Algorithm

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

We illustrate the algorithm with the example having

$$\Upsilon = \{12345, 21345, 23145, 32145, 31245, 13245 \\ 12354, 21354, 23154, 32154, 13254\}$$

and $L = 12345$. **Figure 4** contains the swap graph.

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

The first time that Step 8 is executed in Partial-Cover, $L_1 = 12345$ and $L_2 = 21345$. (There are three choices for L_2. 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 21435 is not in Υ'. The resulting cover relation $(3,4)$ is not new, so A remains $A = \{(3,4)\}$. The **while** loop from Steps 13 to 21 has only the swap pair $\{1,2\}$ to work with. Linear order 32154 is missing its $\{1,2\}$ swap partner, 31254. Hence, 32154 is deleted from Υ', which is now

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

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 23415 is not in Υ'. The resulting cover relation $(1,4)$ is new, so *A* is extended to $A = \{(1,4),(3,4)\}$. None of the linear orders in Υ' has 4 less than 1, so the call to Trim does not change Υ'. 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 Υ', which is now

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

The third time that Step 8 is executed, $L_1 = 21345$ and $L_2 = 21354$. Then $\Lambda = \{12345, 21345, 23145, 21354\}$ (Step 9) and $B = \{\{1,2\},\{1,3\},\{4,5\}\}$ (Step 10). The call to Trim in Step 11 finds that 21534 is not in Υ'. 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 Υ' has 5 less than 3, so the call to Trim does not change Υ'. 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-

```
Partial-Cover(ϒ, L)
1   ▷ Return a poset that is a partial cover for ϒ including L
2   Λ ← {L}
3   Set ϒ' to the set of linear orders in the connected component of G(ϒ) that contains L
4   A ← ∅; B ← ∅
5   (A, ϒ') ← Trim(A, ϒ', Λ)
6   Set ϒ' to the set of linear orders in the connected component of G(ϒ) that contains L
7   while ϒ' ≠ Λ
8         do Select L1 ∈ Λ and L2 ∈ ϒ' \ Λ such that L1 ↔ L2
9            Λ ← Λ ∪ {L2}
10           B ← B ∪ {SwapPair(L1, L2)}
11           (A, ϒ') ← Trim(A, ϒ', Λ)
12           again ← True
13           while again
14                 do again ← False
15                    for L3 ∈ ϒ'
16                        do for {a, b} ∈ B
17                               do if a ≺L3 b or b ≺L3 a
18                                     then L4 ← Swap[L3; {a, b}]
19                                          if L4 ∉ ϒ' then ϒ' ← ϒ' \ {L3}
20                                                         (A, ϒ') ← Trim(A, ϒ', Λ)
21                                                         again ← True
22           Set ϒ' to the set of linear orders in the connected component of G(ϒ')
                 that contains L
23  return (A, B, Λ)
```

Figure 2. Pseudocode for partial-cover (Υ, L).


```
TRIM(A, Υ′, Λ)
    ▷ Ensure that Υ′ ∈ ℒ(min A)
    ▷ and that A contains all order relations implied by Λ and Υ′
    done ← FALSE
    while NOT done
        do done ← TRUE
           for L ∈ Λ
               do for i ← 1 to n − 1
                      do L′ ← Swap[L; i]
                         if L′ ∉ Υ′
                            then A ← A ∪ {(L[i], L[i + 1])}
           for L ∈ Υ′
               do if L ∉ ℒ(Min(A))
                     then Υ′ ← Υ′ \ {L}
                          done ← FALSE
    return (A, Υ′)
```

Figure 3. Pseudocode for trim (A, Υ', Λ).

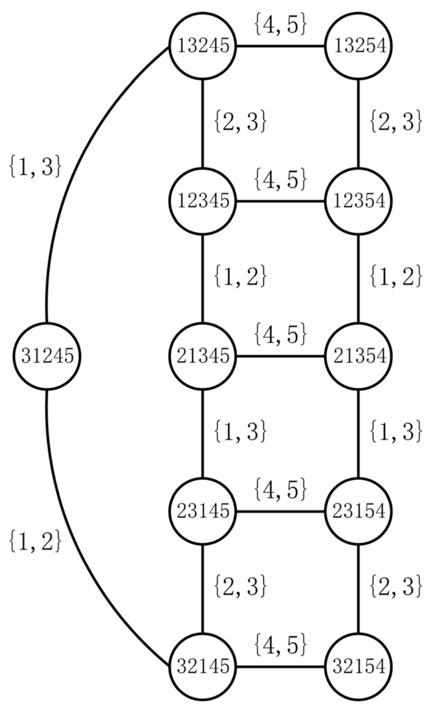

Figure 4. Swap graph for example.

ners. Hence, they are deleted from Υ', which is now

$$\Upsilon' = \{12345, 21345, 23145, 12354, 21354, 23154\}.$$

The fourth time that Step 8 is executed, $L_1 = 21354$ and $L_2 = 12354$. Then $\Lambda = \{12345, 21345, 23145, 21354, 12354\}$ (Step 9) and $B = \{\{1,2\},\{1,3\},\{4,5\}\}$ (Step 10). The call to Trim in Step 11 finds that 13254 and 12534 are not in Υ'. The resulting cover relations are $(2,3)$ and $(3,5)$, so A is extended to $A = \{(1,4),(2,3),(3,4),(3,5)\}$. None of the linear orders in Υ' has 3 less than 2, so the call to Trim does not change Υ'. 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 Υ', which remains

$$\Upsilon' = \{12345, 21345, 23145, 12354, 21354, 23154\}.$$

The fifth and last time that Step 8 is executed, $L_1 = 21354$ and $L_2 = 23154$. Then $\Lambda = \{12345, 21345, 23145, 21354, 12354\}$ (Step 9) and $B = \{\{1,2\},\{1,3\},\{4,5\}\}$ (Step 10). The call to Trim in Step 11 finds that 32154 and 23514 are not in Υ'. 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 Υ' has 3 less than 2 or 5 less than 1, so the call to Trim does not change Υ'. 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 Υ', which remains

$$\Upsilon' = \{12345, 21345, 23145, 12354, 21354, 23154\}.$$

At this point, $\Lambda = \Upsilon'$.

The resulting poset P has the cover relations in A, namely, $1 \prec_P 4, 1 \prec_P 5, 2 \prec_P 3, 3 \prec_P 4$, and $3 \prec_P 5$. The set of linear extensions of P is exactly the final value of Υ', 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) $L \in \Lambda \subseteq \Upsilon' \subseteq \Upsilon$.

2) $\mathcal{G}(\Upsilon')$ 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 Υ' is a linear extension of $\text{Min}(A)$, that is, $\Upsilon' \subseteq \mathcal{L}(\text{Min}(A))$.

5) The set A equals the set of ordered pairs $(a,b) \in X \times X$ for which there exists $L' \in \Lambda$ such that $\text{Swap}[L';\{a,b\}] \notin \Upsilon'$.

6) $\text{Min}(A) \in \text{Down}(B)$ and consequently $\text{Up}(A) \cap \text{Down}(B) \neq \varnothing$.

7) The set B equals the set of unordered pairs $\{a,b\} \subseteq X$ such that $a \neq b$ and such that there exist $L', L'' \in \Lambda$ satisfying $\text{SwapPair}(L', L'') = \{a,b\}$.

8) Let $Q = L_0, L_1, \cdots, L_k$ be a B-labeled path of linear orders such that $L_0 \in \Upsilon', a \prec_{L_0} b,$ and $a \prec_{L_k} b$ and such that Q is a shortest B-labeled path from L_0 to L_k. Let $Q' = L_{0'}, L_{1'}, \cdots, L_{k'}$ be the $\{a,b\}$-mirror path for Q. Then, all of the L_i's are in Υ', and either all of the $L_{i'}$'s are in Υ' 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

5.

Invariant 2 is guaranteed by Steps 6 and 22 and by the way that linear orders are selected for addition to Λ (Step 8).

Invariants 1 and 2 guarantee that, whenever Step 8 is reached, there is a suitable L_1, L_2 pair to select.

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

Initialization. After initialization (Steps 2 through 6), all invariants are true for the first execution of Step 7, for the following reasons. We have $\Lambda = \{L\}$ and $B = \varnothing$. The execution of Trim (Step 5) ensures that Invariants 4 and 5 hold, while maintaining $L \in \Lambda \subseteq \Upsilon'$ (Invariant 1). Step 6 guarantees Invariant 2. Invariant 3 holds because the only order relations in A are cover relations in L. The fact that $B = \varnothing$ makes Invariants 6, 7, and 8 true vacuously.

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

The algorithm never deletes an element of Λ from Υ' 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 Λ deleted. The deletion of L_j is caused by a sequence $a = c_1, c_2, \cdots, c_k = b$ 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 $c_i \prec_{\hat{L}_i} c_{i+1}$, and $\text{Swap}\left[\hat{L}_i; \{c_i, c_{i+1}\}\right] \notin \Upsilon'$. There is a path in $\mathcal{G}(\Lambda)$ 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 $\hat{L}_i$ and in L_j, 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 Λ from Υ' in Step 19. The deletion of an element $L_3 \in \Upsilon'$ depends on the swap pairs in B. More particularly, such a deletion would require a B-labeled path in $\mathcal{G}(\Upsilon')$ labeled from B from L_3 to some L_4 that has a swap pair from B that goes to a linear order outside Υ'. This cannot happen because of Invariant 8. We conclude that L_3 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 $P = L_0, L_1, \cdots, L_k$ be a B-labeled path of linear orders such that $L_0 \in \Upsilon'$, $a \prec_{L_0} b$, and $a \prec_{L_k} b$ and such that P is a shortest B-labeled path from L_0 to L_k. Let $P' = L_{0'}, L_{1'}, \cdots, L_{k'}$ be the $\{a,b\}$-mirror path for P.

To obtain a contradiction, assume that there is some L_i that is not in Υ. Let L_i be the first such. Then $i \neq 0$, so $L_{i-1} \in \Upsilon$. Let $\{c,d\} = \text{SwapPair}(L_{i-1}, L_i) \in B$. Since $L_i \notin \Upsilon$, L_{i-1} cannot be in Υ, 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_i's are in Υ.

We next show that P is not just a shortest B-labeled path but is also a shortest path in $\mathcal{G}(\mathcal{L}(\varnothing))$. Let $C = \left\{\{c,d\} \middle| c <_{L_0} d \text{ and } d <_{L_k} c\right\}$. For any path from L_0 to L_k in $\mathcal{G}(\mathcal{L}(\varnothing))$, 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 $\mathcal{G}(\mathcal{L}(\varnothing))$ 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 $a \prec_{L_0} b$ and $a \prec_{L_k} b$, 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 $\{c,d\} \cap \{a,b\} = \varnothing$, 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 $L_{i'}$'s are in Υ' or none of them are. To obtain a contradiction, suppose that $L_{i'} \in \Upsilon'$ and that $L'_{i+1} \notin \Upsilon'$, for some i satisfying $0 \leq i \leq k-1$. (The case $L_{i'} \notin \Upsilon'$ and $L'_{i+1} \in \Upsilon'$ will yield a contradiction by an analogous argument.) But, in this case, $L_{i'}$ would have been deleted from Υ' in an earlier iteration, a contradiction. From this contradiction, we conclude that either all of the $L_{i'}$'s are in Υ' 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(Υ, L) *returns a set* A *such that* $\mathrm{Min}(A)$ *is a partial cover of* Υ *including* L. *The algorithm has time complexity* $O(n^2|\Upsilon|^2)$.

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

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

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

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 $O(n^2|\Upsilon|^2)$, as required.

REFERENCES

[1] B. S. Baker and E. G. Coffman, "Mutual Exclusion Scheduling," *Theoretical Computer Science*, Vol. 162, No. 2, 1996, pp. 225-243. doi:10.1016/0304-3975(96)00031-X

[2] P. Barcia and J. O. Cerdeira, "The *k*-Track Assignment Problem on Partial Orders," *Journal of Scheduling*, Vol. 8, No. 2, 2005, pp. 135-143. doi:10.1007/s10951-005-6363-6

[3] F. A. Chudak and D. B. Shmoys, "Approximation Algorithms for Precedence-Constrained Scheduling Problems on Parallel Machines That Run at Different Speeds," *Journal of Algorithms*, Vol. 30, No. 2, 1999, pp. 323-343. doi:10.1006/jagm.1998.0987

[4] J. R. Correa and A. S. Schulz, "Single-Machine Scheduling with Precedence Constraints," *Mathematics of Operations Research*, Vol. 30, No. 4, 2005, pp. 1005-1021. doi:10.1287/moor.1050.0158

[5] T. Kis, "Job-Shop Scheduling with Processing Alternatives," *European Journal of Operational Research*, Vol. 151, No. 2, 2003, pp. 307-332. doi:10.1016/S0377-2217(02)00828-7

[6] M. Peter and G. Wambach, "*n*-Extendible Posets, and How to Minimize Total Weighted Completion Time," *Discrete Applied Mathematics*, Vol. 99, No. 1-3, 2000, pp. 157-167. doi:10.1016/S0166-218X(99)00131-6

[7] N. Policella, A. Oddi, S. F. Smith and A. Cesta, "Generating Robust Partial Order Schedules," *Proceedings of Principles and Practice of Constraint Programming*, Vol. 3258, 2004, pp. 496-511.

[8] D. B. Shmoys, C. Stein and J. Wein, "Improved Approximation Algorithms for Shop Scheduling Problems," *SIAM Journal on Computing*, Vol. 23, No. 3, 1994, pp. 617-632. doi:10.1137/S009753979222676X

[9] C. Grasso and C. Lee, "Combining Partial Order Alignment and Progressive Multiple Sequence Alignment Increases Alignment Speed and Scalability to Very Large Alignment Problems," *Bioinformatics*, Vol. 20, No. 10, 2004, pp. 1546-1556. doi:10.1093/bioinformatics/bth126

[10] S. K. Kannan and T. J. Warnow, "Tree Reconstruction from Partial Orders," *SIAM Journal on Computing*, Vol. 24, No. 3, 1995, pp. 511-519. doi:10.1137/S0097539793252195

[11] S. Karlin and I. Ladunga, "Comparisons of Eukaryotic Genomic Sequences," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 91, No. 26, 1994, pp. 12832-12836. doi:10.1073/pnas.91.26.12832

[12] K. Miyakawa and H. Narushima, "Lattice-Theoretic Properties of MPR-Posets in Phylogeny," *Discrete Applied Mathematics*, Vol. 134, No. 1-3, 2004, pp. 169-192. doi:10.1016/S0166-218X(03)00303-2

[13] P. L. Hammer, A. Kogan and M. A. Lejeune, "Modeling Country Risk Ratings Using Partial Orders," *European Journal of Operational Research*, Vol. 175, No. 2, 2006, pp. 836-859. doi:10.1016/j.ejor.2005.06.040

[14] S. Holland and W. Kiessling, "User Preference Mining Techniques for Personalized Applications," *Wirtschaftsinformatik*, Vol. 46, No. 6, 2004, pp. 439-445. doi:10.1007/BF03250961

[15] S. Holland, M. Ester and W. Kiessling, "Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications," *Proceedings of Knowledge Discovery in Databases*: *PKDD* 2003, Vol. 2838, 2003, pp. 204-216.

[16] H. Mannila, H. Toivonen and A. I. Verkamo, "Discovery of Frequent Episodes in Event Sequences," *Data Mining and Knowledge Discovery*, Vol. 1, No. 3, 1997, pp. 259-289. doi:10.1023/A:1009748302351

[17] J. Pei, H. X. Wang, J. Liu, K. Wang, J. Y. Wang, and P. S. Yu, "Discovering Frequent Closed Partial Orders from Strings," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 11, 2006, pp. 1467-1481. doi:10.1109/TKDE.2006.172

[18] B. Bollobas and G. Brightwell, "The Structure of Random Graph Orders," *SIAM Journal on Discrete Mathematics*, Vol. 10, No. 2, 1997, pp. 318-335. doi:10.1137/S0895480194281215

[19] S. Felsner and K. Reuter, "The Linear Extension Diameter of a Poset," *SIAM Journal on Discrete Mathematics*,

Vol. 12, No. 3, 1999, pp. 360-373. doi:10.1137/S0895480197326139

[20] S. Felsner and W. T. Trotter, "Posets and Planar Graphs," *Journal of Graph Theory*, Vol. 49, No. 4, 2005, pp. 273-284. doi:10.1002/jgt.20081

[21] P. C. Fishburn, P. J. Tanenbaum and A. N. Trenk, "Linear Discrepancy and Bandwidth," *Order*, Vol. 18, No. 3, 2001, pp. 237-245. doi:10.1023/A:1012267732204

[22] L. S. Heath and S. V. Pemmaraju, "Stack and Queue Layouts of Posets," *SIAM Journal on Discrete Mathematics*, Vol. 10, No. 4, 1997, pp. 599-625. doi:10.1137/S0895480193252380

[23] M. Naatz, "The Graph of Linear Extensions Revisited," *SIAM Journal on Discrete Mathematics*, Vol. 13, No. 3, 2000, pp. 354-369. doi:10.1137/S0895480199352609

[24] G. Agnarsson, S. Felsner and W. T. Trotter, "The Maximum Number of Edges in a Graph of Bounded Dimension, with Applications to Ring Theory," *Discrete Mathematics*, Vol. 201, No. 1-3, 1999, pp. 5-19. doi:10.1016/S0012-365X(98)00309-4

[25] M. Aguiar and W. F. Santos, "Galois Connections for Incidence Hopf Algebras of Partially Ordered Sets," *Advances in Mathematics*, Vol. 151, No. 1, 2000, pp. 71-100. doi:10.1006/aima.1999.1864

[26] N. Bergeron and F. Sottile, "Hopf Algebras and Edge-Labeled Posets," *Journal of Algebra*, Vol. 216, No. 2, 1999, pp. 641-651. doi:10.1006/jabr.1998.7794

[27] J. Konieczny, "Reduced Idempotents in the Semigroup of Boolean Matrices," *Journal of Symbolic Computation*, Vol. 20, No. 4, 1995, pp. 471-482. doi:10.1006/jsco.1995.1059

[28] F. Ruskey, "Generating Linear Extensions of Posets by Transpositions," *Journal of Combinatorial Theory Series B*, Vol. 54, No. 1, 1992, pp. 77-101. doi:10.1016/0095-8956(92)90067-8

[29] D. B. West, "Generating Linear Extensions by Adjacent Transpositions," *Journal of Combinatorial Theory Series B*, Vol. 58, No. 1, 1993, pp. 58-64. doi:10.1006/jctb.1993.1031

[30] G. Pruesse and F. Ruskey, "Generating Linear Extensions Fast," *SIAM Journal on Computing*, Vol. 23, No. 2, 1994, pp. 373-386. doi:10.1137/S0097539791202647

[31] E. R. Canfield and S. G. Williamson, "A Loop-Free Algorithm for Generating the Linear Extensions of a Poset," *Order*, Vol. 12, No. 1, 1995, pp. 57-75.

[32] J. F. Korsh and P. S. LaFollette, "Loopless Generation of Linear Extensions of a Poset," *Order*, Vol. 19, No. 2, 2002, pp. 115-126. doi:10.1023/A:1016548222238

[33] A. Ono and S. Nakano, "Constant Time Generation of Linear Extensions," *Proceedings of Fundamentals of Computational Theory*, Vol. 3623, 2005, pp. 445-453.

[34] R. Bubley and M. Dyer, "Faster Random Generation of Linear Extensions," *Discrete Mathematics*, Vol. 201, No. 1-3, 1999, pp. 81-88. doi:10.1016/S0012-365X(98)00333-1

[35] P. L. Fernandez, L. S. Heath, N. Ramakrishnan, M. Tan and J. P. C. Vergara, "Mining Posets from Linear Orders," *Discrete Mathematics, Algorithms and Applications*, 2013, in press.

[36] P. L. Fernandez, L. S. Heath, N. Ramakrishnan and J. P. C. Vergara, "Reconstructing Partial Orders from Linear Extensions," *Proceedings of the Fourth SIGKDD Workshop on Temporal Data Mining*: *Network Reconstruction from Dynamic Data*, Philadelphia, 20 August 2006, p. 4.

[37] A. K. Lee and M. A. Wilson, "A Combinatorial Method for Analyzing Sequential Firing Patterns Involving an Arbitrary Number of Neurons Based on Relative Time Order," *Journal of Neurophysiology*, Vol. 92, No. 4, 2004, pp. 2555-2573. doi:10.1152/jn.01030.2003

[38] B. A. Davey and H. A. Priestley, "Introduction to Lattices and Order," 2nd Edition, Cambridge University Press, Cambridge, 2002. doi:10.1017/CBO9780511809088

[39] G. Brightwell and P. Winkler, "Counting Linear Extensions," *Order*, Vol. 8, No. 3, 1991, pp. 225-242. doi:10.1007/BF00383444

[40] M. R. Garey, D. S. Johnson and L. Stockmeyer, "Some Simplified *NP*-Complete Graph problems," *Theoretical Computer Science*, Vol. 1, No. 3, 1976, pp. 237-267. doi:10.1016/0304-3975(76)90059-1

[41] T. M. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms," 2nd Edition, The MIT Press, Cambridge, 2001.

