# On Finding Tucker Submatrices and Lekkerkerker-Boland Subgraphs

Nathan Lindzey *, Ross M. McConnell **

Colorado State University, Fort Collins CO 80521, USA

**Abstract.** Lekkerkerker and Boland characterized the minimal forbidden induced subgraphs for the class of interval graphs. We give a linear-time algorithm to find one in any graph that is not an interval graph. Tucker characterized the minimal forbidden submatrices of matrices that do not have the consecutive-ones property. We give a linear-time algorithm to find one in any matrix that does not have the consecutive-ones property.

## 1 Introduction

A graph is an *interval graph* if it is the intersection graph of a set of intervals on a line. Such a set of intervals is known as an *interval model* of the graph. They are an important subclass of perfect graphs [5], they have been written extensively about and they model constraints in various combinatorial optimization and decision problems [13, 15]. They have a rich structure and history, and interesting relationships to other graph classes. For a survey, see [2].

If $M$ is a 0-1 (binary) matrix, we let $size(M)$ denote the number of rows, columns and 1's. Such a matrix has the *consecutive-ones property* if there exists a reordering of its columns such that, in every row, the 1's are consecutive. A *consecutive-ones matrix* is a matrix that has the consecutive-ones property, and a *consecutive-ones-ordered matrix* is a matrix where the 1's are consecutive in every row. A *clique matrix* of a graph is a matrix that has a row for each vertex, a column for each clique, and a 1 in row $i$, column $j$ if vertex $i$ is contained in clique $j$. A graph is an interval graph if and only if its clique matrices have the consecutive-ones property, see, for example [5].

In 1962, Lekkerkerker and Boland described the minimal induced forbidden subgraphs for the class of interval graphs [8], known as the *LB graphs* (Figure 2). Ten years later, Tucker described the minimal forbidden submatrices for consecutive-ones matrices [19]. These are depicted in Figure 1. Not surprisingly, there is a relationship between the intersection graphs of rows of Tucker matrices and the LB graphs, depicted in Figure 2.

* lindzey@cs.colostate.edu, Computer Science Department, Colorado State University, Fort Collins, CO, 80523-1873 U.S.A.

** rmm@cs.colostate.edu, Computer Science Department, Colorado State University, Fort Collins, CO, 80523-1873 U.S.A.

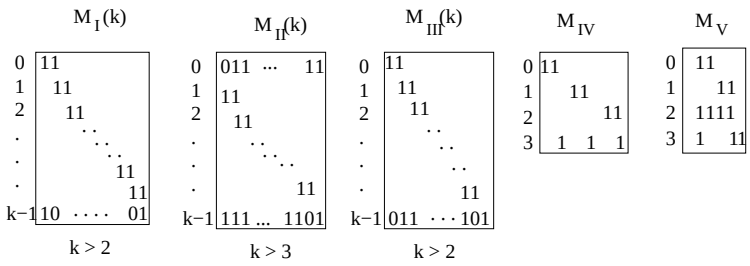


**Fig. 1.** The minimal forbidden submatrices for consecutive-ones matrices. For $M_I$, $k \geq 3$, and for $M_{II}$ and $M_{II}$, $k \geq 4$. $M_{IV}$ and $M_I$ have fixed size.

In this paper, we give a linear time bound for finding one of the LB subgraphs when a graph is not an interval graph. As part of our algorithm, we also give a linear-time ($O(size(M))$ bound for finding one of Tucker's submatrices in a matrix $M$ that does not have the consecutive-ones property. This latter problem was solved previously in $O(n * size(M))$ time in [17], where $n$ is the number of rows of the matrix. An $O(\Delta^3 m^2 n(m + n^2)$ bound for finding a Tucker of minimum size is given in [4], where $\Delta$ is the maximum number of 1's in any row.

A graph is *chordal* if it has no *chordless cycle* (an induced cycle on four or more vertices). A vertex is *simplicial* if it and its neighbors induce a complete subgraph. Every chordal graph has a simplicial vertex, and every interval graph is chordal [5].

An interval graph is *proper* if there exists an interval model where no interval is a subset of another. It is a *unit* interval graph if there exists an interval model where all intervals have the same length. These graph classes are the same, and Wegner showed that a graph is a proper interval graph if and only if it does not have a chordless cycle, the special case of $G_{IV}$ or $G_V$ for $n = 6$ or the *claw* ($K_{1,3}$) as an induced subgraph [20]. Hell and Huang give an algorithm that produces one of them in linear time [6] . The problem of finding a forbidden subgraph reduces easily to finding an LB subgraph. Each of the LB graphs is either one of Wegner's forbidden subgraphs or contains an obvious claw, and finding a claw in linear time, given an interval model, is elementary. By itself, this approach has no obvious advantages over Hell and Huang's elegant algorithm, but such reductions are useful when studying or programming a collection of related algorithms.

A *certifying algorithm* is an algorithm that provides, with each output, a simple-to-check proof that it has answered correctly [7, 11]. An interval model gives a certificate that a graph is an interval graph, and an LB subgraph gives one if the graph is not an interval graph. However, a certifying algorithm was given previously in [7]. The ability to give a consecutive-ones ordering or a Tucker submatrix in linear time gives a linear-time certifying algorithm for consecutive-ones matrices, but one was given previously in [10]. The previous certificates are easier to check, which is a desirable property for certifying algorithms. However, they are neither minimal nor uniquely characterized. Aside from the theoretical

interest in LB subgraphs, it is easy to obtain a minimal certificate of the form given in [7] from an LB subgraph found by the algorithm we describe below. An open question is whether a minimal, unique, especially simple, or otherwise interesting special case of the certificate from [10] can be obtained by applying the algorithm of that paper to a Tucker submatrix obtained by the algorithm of the present paper. Tucker submatrices may be useful in heuristics for finding large submatrices that have the consecutive-ones property, small Tucker matrices, or identifying errors in biological data [16, 3]. Our techniques provide new tools for such heuristics.

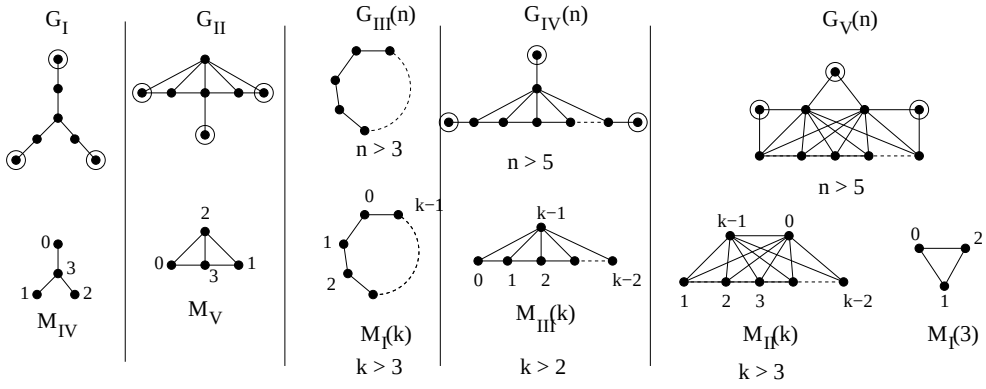


**Fig. 2.** $G_I$ through $G_V$ are the minimal non-interval graphs discovered by Lekkerkerker and Boland. The circled vertices are the simplicial vertices in the graphs. Below them are the intersection graphs of the rows of the corresponding Tucker matrices, numbered as they are in Figure 1. Removing the rows belonging to the simplicial vertices in the clique matrix of each Lekkerkerker-Boland graph gives the Tucker matrix below it in the figure.

## 2 Preliminaries

Given a graph $G$, let $V$ denote the number of vertices and $E$ denote the number of edges. If $\emptyset \subset X \subseteq V$, let $G[X]$ denote the subgraph induced by $X$. Standard sparse representations of 0-1 matrices take $O(size(M))$ space to represent $M$. We treat the rows and columns as *sets*, where each row $R$ is the set of columns in which the row has a 1 and each column $C$ is the set of rows in which the column has a 1. Suppose $\mathcal{R}$ is the set of rows of a consecutive-ones ordered matrix and $(C_1, C_2, \ldots, C_m)$ is the ordering of the columns. In linear time, we can find, for each row, the leftmost and rightmost column in the row. Let us call these the *left endpoint* and *right endpoint* of the row.

That interval graphs are a subclass of the class of chordal graphs follows from inclusion of the $G_{III}$'s among the LB subgraphs. Rose, Tarjan and Lueker

give an $O(V+E)$ algorithm that recognizes whether a graph is a chordal graph, and, if so, produces its maximal cliques [14]. Otherwise, the algorithm of [18] produces a chordless cycle ($G_{III}$) in linear time.

When a graph is chordal, the problem of deciding whether it is an interval graph reduces to the problem of deciding whether its clique matrix has the consecutive-ones property. Booth and Lueker further reduced this problem to that of finding a maximal prefix $\mathcal{R}' = \{R_1, R_2, \ldots, R_r\}$ of the rows of a binary matrix $M$ that has the consecutive-ones property, and give an algorithm that solves this in $O(size(M))$ time [1].

Assigning a left-to-right order to children of each internal node of a rooted tree results in a unique left-to-right order of the leaves. Booth and Lueker's algorithm produces a *PQ tree*, for $\mathcal{R}'$. The PQ tree represents all possible consecutive-ones orderings of $\mathcal{R}'$. There is one leaf $\{c\}$ for each column $c$. The internal nodes of the PQ tree consist of *P nodes* and *Q nodes*. The consecutive-ones ordering of columns are given by the leaf orders obtainable by assigning an arbitrary left-to-right order to children of each P node, and for each Q node, assigning the given left-to-right order or its reverse.

Though the PQ tree can be represented using $O(1)$ space per node, conceptually, we will consider each node of the PQ tree to be a set given by the disjoint union of its children; equivalently, it is the union of its leaf descendants.

**Definition 1.** *Let $\mathcal{S}$ be a collection of subsets of a set $U$. Two elements of $U$ are in the same* Venn class *if they are elements of the same set of members of $\mathcal{S}$. The* unconstrained Venn class *consists of those elements of $U$ that are not in any member of $\mathcal{S}$; all others are* constrained. *Two sets $R_1, R_2$* overlap *if their intersection is nonempty, but neither is a subset of the other. The* overlap graph *of $\mathcal{S}$ is the undirected graph whose vertices are the members of $\mathcal{S}$, and $R_1, R_2 \in \mathcal{S}$ are adjacent if and only if $R_1$ and $R_2$ overlap.*

**Lemma 1.** *[12] A set of columns is a Q node of a consecutive-ones matrix $M$ if and only if it is the union of rows of a connected component of the overlap graph of rows of $M$. The Venn classes of rows in this component are its children.*

## 3 Breadth-first search on the overlap graph of a collection of sets, given a consecutive-ones ordering

In linear time, we may label each row of a consecutive-ones ordered matrix with its left and right endpoints. We may then label each column $c_i$ of a consecutive-ones ordered matrix with the set of rows that have their left endpoints in $c_i$. In linear time, we can then radix sort the list of sets that have their left endpoint at $c_i$ in descending order of index of right endpoint, yielding a list $\mathcal{R}_i$. This is accomplished with a single radix sort that has the index of the left endpoint as its primary sort key and index of the right endpoint as the secondary sort key. By symmetry, we can construct a list $\mathcal{L}_i$ of rows that have their right endpoint in each column $c_i$, sorted in ascending order of index of left endpoint.

This allows us to perform a breadth-first search on the overlap graph of the rows in time linear in the size of the matrix, as follows. The lists $\mathcal{L}_i$ and $\mathcal{R}_i$ are represented with doubly-linked lists. We maintain the invariant that elements that have been placed in the BFS queue have been removed from these lists. When a consecutive-ones ordered set $R$ comes to the front of the queue, we traverse its list $(c_j, c_{j+1}, \ldots, c_k)$ of columns. For each $c_h$ in the list, we remove elements from $\mathcal{R}_h$ and place them in the queue, until we reach an element in $\mathcal{R}_i$ whose right endpoint is no farther to the right than $c_k$. All of the removed elements overlap $R$. Since $\mathcal{R}_h$ is sorted in descending order of right endpoint, all these elements are a prefix of $\mathcal{R}_h$, and any remaining elements in the list do not overlap $R$. When we remove an element from $\mathcal{R}_h$, we remove it from any list $\mathcal{L}_{h'}$ that it is a member of, to maintain the invariant. This takes $O(1)$ time for each element moved to the BFS queue, plus $O(1)$ time for each column of $R$. The lists $L_h$ are handled symmetrically. Summing over all rows $R$, the time is $O(size(M))$.

## 4 Finding Tucker Submatrices

### 4.1 Tucker matrices with at most four rows

**Lemma 2.** *If a set $\mathcal{R}'$ of rows has the consecutive-ones property and $Z$ is a row such that $\mathcal{R} = \mathcal{R}' \cup \{Z\}$ does not, then $Z$ is one of the rows of every Tucker submatrix in $\mathcal{R}$.*

**Algorithm 1** – *Preconditions: $M'$ does not have the consecutive-ones property and $k \geq 1$.*
– *Postconditions are given by Lemma 3.*

*initialRows($M', k$)*
  $M := M'$
  $i := 1$;
  *While $i \leq k+1$ and $M$ has at least $i-1$ rows*
    *Using Booth and Lueker's algorithm [1], find the minimal prefix $(R_1, R_2, \ldots, R_r, Z)$ of rows of $M$ that does not have the consecutive-ones property.*
    *Let $M$ be the matrix whose row sequence is $(Z, R_1, R_2, \ldots, R_r)$.*
    *i := i+1*
  *return M*

**Lemma 3.** *Suppose Algorithm 1 is run with parameter $k$ and a matrix $M'$ that does not have the consecutive-ones property. If the returned matrix $M$ has at most $k$ rows, then these are the rows of every Tucker matrix of $M$. Otherwise, $M$ fails to have the consecutive-ones property and every Tucker submatrix in $M$ has at least $k+1$ rows.*

*Proof.* By induction on $i$, $M$ does not have the consecutive-ones property at the end of iteration $i$. Also, by induction on $i$, using Lemma 2, at the end of iteration $i$, for every Tucker submatrix $M_T$ of $M$, the rows of $M_T$ include the first $i$ rows of $M$. If $M_T$ has only $i$ rows, then the first $i$ rows of $M$ do not have the consecutive-ones property, so at the end of iteration $i+1$, $M$ will have $i$ rows.

We run Algorithm 1 for $k = 4$. If it returns a matrix with $j$ rows, where $j \leq 4$, it is easy to get a linear time bound to get the columns. (One way is to generate all $j! \leq 24$ orderings of rows and for each, to check for the columns of each Tucker matrix of size $j$.) Otherwise, Algorithm 1 returns a matrix $M$ of more than 4 rows. By Lemma 3, $M$ fails to have the consecutive-ones property and every Tucker submatrix of $M$ has at least five rows. This excludes any instances of $M_{IV}$, $M_V$ or the anomalous case of $M_I$ on three rows that does not correspond to a chordless cycle.

### 4.2 Matrices in which all Tucker submatrices have more than four rows

**Lemma 4.** *The overlap graphs of $M_I$, $M_{II}$, and $M_{III}$ are simple cycles.*

**Definition 2.** *Suppose $\mathcal{R}'$ is a set of rows with the consecutive-ones property, $Q$ is a $Q$ node of its PQ tree, $(X_1, X_2, \ldots, X_k)$ is the ordering of $Q$'s children and $Z$ is a row not in $\mathcal{R}'$. Let $X_h, X_i, X_j$ be three children of $Q$ such that $h < i < j$. They are a* 1-0-1 configuration for $Z$ *if $X_h$ and $X_j$ each contain a 1 of row $Z$ and $X_i$ contains a 0 of row $Z$. They are a* 0-1-0 configuration *for $Z$ if $X_h$ and $X_j$ each contain a 0 of row $Z$ and $X_i$ contains a 1 of row $Z$.*

**Lemma 5.** *If $\mathcal{R}'$ is a set of rows that has the consecutive-ones property and $Z$ is a row not in $\mathcal{R}'$, then $\mathcal{R}' \cup \{Z\}$ does not have the consecutive-ones property if the PQ tree of $\mathcal{R}'$ has a $Q$ node $Q$ such that either:*

1. *$Q$ has a 1-0-1 configuration for $Z$;*
2. *$Q$ has a 0-1-0 configuration for $Z$ and $Z$ is not a subset of $Q$.*

This test is implicit in Booth and Lueker's algorithm, where it is a sufficient condition, but not a necessary one. The following is a consequence of Lemma 1.

**Lemma 6.** *[10] The conditions of Lemma 5 are necessary and sufficient if the overlap graph of $\mathcal{R}'$ is connected.*

**Lemma 7.** *If a matrix fails to have the consecutive-ones property and has no Tucker submatrix with fewer than five rows, then when Algorithm 1 is run on it with $k = 4$, at the end of one of the five iterations of its loop, the PQ tree of $\mathcal{R}' = \{R_1, R_2, \ldots, R_r\}$ will have a $Q$ node $Q$ with the following properties for the row $Z$ of the iteration:*

- *$Q$ has a 1-0-1 configuration $(X_h, X_i, X_j)$ for $Z$;*

– *There exist $A, B \in \mathcal{R}'$ that are members of the component of the overlap graph on $\mathcal{R}'$ whose union is $Q$, and such that $A$ contains $X_h$ and is disjoint from $X_i$ and $X_j$, and $B$ contains $X_j$ and is disjoint from $X_h$ and $X_i$.*

*Proof.* If $\mathcal{T}$ is the rows of $M$ that contain a Tucker submatrix $M_T$, at the end of an iteration of the loop of Algorithm 1, $Z \in \mathcal{T}$ by by Lemma 2. By Lemma 4, the overlap graph of $\mathcal{T}' = \mathcal{T} \setminus \{Z\}$ is connected, so $\mathcal{T}'$ is a subset of a component of the overlap graph of $\mathcal{R}'$, which gives rise to a Q node $Q$ of the PQ tree of $\mathcal{R}'$, by Lemma 1. Since the children of $Q$ are the Venn classes of the component, no two Venn classes of $\mathcal{T}'$, hence no two columns of $M_T$, can lie in the same child of the Q node.

For each choice of a last row of a Tucker matrix on at least five rows, Figure 3 gives the possible orderings imposed on the last row by a consecutive-ones ordering of $\mathcal{T}'$, which is unique up to reversal, by Lemmas 4 and 1. In each case, if row $i \notin \{0, 1, k-2, k-1\}$ is chosen to go last, rows $i-1$ and $i+1$ satisfy the requirements of $A$ and $B$. $M_T$ has at least five rows and no row of $M_T$ is contained in $Z$ more than once in the five iterations, so in at least one of the iterations a row $i \notin \{0, 1, k-2, k-1\}$ will go last.

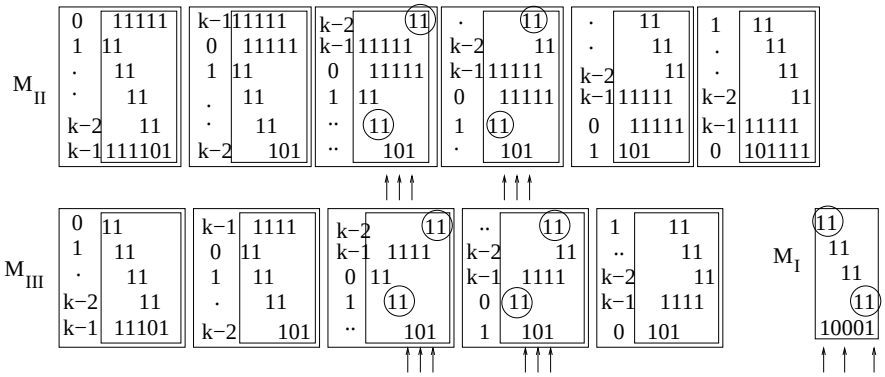


**Fig. 3.** Consecutive-ones orderings of all but the last row of $M_I$, $M_{II}$ and $M_{III}$ for different choices of the last row. For all but at most four choices of the last row, there exists a 1-0-1 configuration and rows $A$ and $B$ satisfying Lemma 7.

In $O(size(M))$ time, all nodes can be labeled as having no descendants in $Z$ (*empty*), all descendants in $Z$ (*full*), or some descendants in $Z$ and some not (*partial*), working from leaves toward the root. A procedure is given in Booth and Lueker's paper. Given this labeling, checking for the conditions of Lemma 7 in $O(size(M))$ time is trivial; details are omitted.

The correctness and linear time bound for the following are the key results of this section:

**Algorithm 2** – *Preconditions: $M$ does not have the consecutive-ones property or any Tucker matrix with fewer than five rows.*
– *Postconditions: The rows of a Tucker matrix of $M$ have been returned.*

*findRows($M$)*
*Run initialRows($M, 4$) (Algorithm 1) to find $\mathcal{R}'$, $Z$, $A$, $B$ satisfying the requirements of Lemma 7.*
*Let $P$ be a shortest path in the overlap graph of $\mathcal{R}'$ from $A$ to $B$ and let $\mathcal{P}$ be its rows.*
*Let $P_1$ be a minimal prefix of $P$ such that the union of $\{Z\}$ and the set $\mathcal{P}_1$ of rows of $P_1$ does not have the consecutive-ones property.*
*Let $P_2$ be a minimal suffix of $P_1$ such that the union of $\{Z\}$ and the set $\mathcal{P}_2$ of rows of $P_2$ does not have the consecutive-ones property.*
*Return $\mathcal{P}_2 \cup \{Z\}$.*

**Lemma 8.** *If $M$ does not have the consecutive-ones property and every Tucker matrix of $M$ has at least five rows, then Algorithm 2 returns the set of rows of a Tucker matrix of $M$.*

*Proof.* Since $A$ and $B$ lie in the same component of the overlap graph of $\mathcal{R}'$, $P$ exists. Since $\mathcal{R}'$ has the consecutive-ones property, so does $\mathcal{P}$. Because $\mathcal{P}$ has a connected overlap graph, $P$, $\bigcup\mathcal{P}$ is a single Q node of the PQ tree of $\mathcal{P}$, by Lemma 1. Because of $A$ and $B$, $X_h$, $X_i$, and $X_j$ are contained in distinct Venn classes of $\mathcal{P}$, and the ones containing $X_h$ and $X_j$ are constrained. Since $\bigcup\mathcal{P}$ is consecutive, it must have a row that contains $X_i$, hence the Venn class of $\mathcal{P}$ containing $X_i$ is also constrained. Therefore, $\mathcal{P} \cup \{Z\}$ does not have the consecutive-ones property by Lemma 5, and $P_1$ and $P_2$ exist. By Lemma 2, all Tucker matrices in $\mathcal{R}' \cup \{Z\}$ contain $Z$, so this applies also to $\mathcal{P}_2 \cup \{Z\}$.

Suppose there is a proper subset $\mathcal{R}''$ of the rows on $P_2$ such that $\mathcal{R}'' \cup \{Z\}$ contains a Tucker matrix. The overlap graph of $\mathcal{R}''$ is connected, by Lemma 4. Since $P_2$ is a shortest path, it is a chordless path, so $\mathcal{R}''$ is a subpath of $P_2$ by Lemma 4. Let $\mathcal{R}'_1$ be the rows on $P_1$, excluding the last row on $P_1$. Let $\mathcal{R}'_2$ be the rows of $P_2$, excluding the first row on $P_2$. By the minimality of $P_1$ and $P_2$, $\mathcal{R}'_1 \cup \{Z\}$ and $\mathcal{R}'_2 \cup \{Z\}$ have the consecutive-ones property. Since $\mathcal{R}''$ is a subpath of $P_2$, $\mathcal{R}'' \subseteq \mathcal{R}'_1$ or $\mathcal{R}'' \subseteq \mathcal{R}'_2$, so $\mathcal{R}'' \cup \{Z\}$ has the consecutive-ones property, contradicting our assumption that it does not. Therefore, $\mathcal{P}_2 \cup \{Z\}$ is the set of rows of a Tucker matrix.

Figure 4 gives an example on which we illustrate some implementation details. $Z$ is given by the 1's and 0's above the column numbers in the figure on the left, and $\mathcal{R}'$ is depicted by the intervals. The rows labeled $A$ and $B$ satisfy the requirements of $A$ and $B$ for Lemma 7, and $P = (A, E, F, G, H, J, L, B)$ is a shortest path from $A$ to $B$ in the overlap graph of $\mathcal{R}'$, found using the BFS algorithm of Section 3.

Using Booth and Lueker's terminology, we maintain labels on each class indicating whether it is *full* (contains only 1's of $Z$), *empty* (contains only 0's of $Z$), or *partial* (contains both 1's and 0's of $Z$). The minimal prefix $P_1$ of $P$ whose rows,

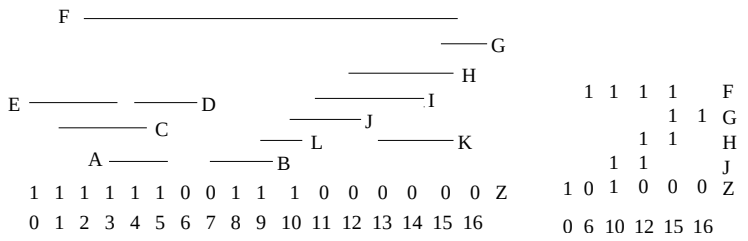


**Fig. 4.** Example of finding a minimal set of rows that does not have the consecutive-ones property

together with $Z$, do not have the consecutive-ones property, is $(A, E, F, G, H, J)$. This is detected as follows. It is easy to verify that its sequence of constrained Venn classes is ($\{0, 1\}$,$\{2\}$,$\{3\}$, $\{4, 5\}$, $\{6, 7, \ldots, 9\}$, $\{10, 11\}$, $\{12\}$, $\{13, 14\}$, $\{15\}$, $\{16\}$), and their full/partial/empty labels $(F, F, F, F, P, P, E, E, E, E)$, respectively. Selecting a 1 from a full class, a 0 from the first partial class and a 1 from the second partial class gives a 1-0-1 configuration satisfying Lemma 5. It is the minimal such prefix. A smaller prefix, $(A, E, F)$ has a 0-1-0 configuration, but it does not satisfy Lemma 5 because $Z \subset A \cup E \cup F$.

The minimal suffix $P_2$ of $P_1$ that satisfies Lemma 5 is $(F, G, H, J)$, which is found in the same way by working on the reverse of $P_1$. Its sequence of constrained Venn classes are ($\{2, 3, \ldots, 9\}$, $\{10, 11\}$, $\{12 - 14\}$, $\{15\}$, $\{16\}$), labeled $(P, P, E, E, E)$, respectively. Selecting a 0 from the first partial class, a 1 from the next, and a 0 from an empty class gives a 0-1-0 configuration. It satisfies condition 2 of the lemma, because the unconstrained class, $\{0, 1\}$ is partial, hence $Z \not\subseteq F \cup G \cup H \cup J$.

Therefore, $\{F, G, H, J, Z\}$ is the set of rows of a Tucker submatrix. A minimal set of columns that illustrates that it satisfies the lemma is $\{0, 6, 10, 12, 15, 16\}$. On the righthand side of Figure 4 is the resulting Tucker matrix, which matches the final configuration in the sequence for $M_{III}$ in Figure 3.

This example shows that the key to finding $P_1$ and $P_2$ is maintaining the sequence of constrained Venn classes and their full/partial/empty labels as rows are added in the order in which they occur on $P$ or on the reverse of $P_1$. Since they are added in an order such that every prefix of the order has a connected overlap graph, the sequence is uniquely constrained after each row is added, by Lemma 1. When a row $R_i$ is added, if it overlaps a constrained Venn class $X$, $X$ must be replaced in the sequence with two Venn classes, $(X \setminus R_i, X \cap R_i)$ or with $(X \cap R_i, X \setminus R_i)$, whichever is required to maintain consecutiveness of $R_i$. If $R_i$ intersects the unconstrained class, $S$, then $R_i \cap S$ must be added at one extreme end of the sequence, whichever maintains consecutiveness of $R_i$. Details are given in [10].

The difference between this algorithm and that of [10] (and Booth and Lueker) is that, instead of testing at each iteration whether the next row $R_i$ can be added to those considered so far without undermining the consecutive-ones property, it must repeatedly perform this test on the fixed row $Z$ after

each row $R_i$ is added. We already know that $R_i$ can be added, since $\mathcal{R}'$ has the consecutive-ones property. Like Booth and Lueker, the previous algorithm of [10] applies the full/partial/empty labels for $R_i$ to facilitate the test, in $O(|R_i|)$ time, and then removes them before considering the next row $R_{i+1}$. Though we must perform the test on the fixed row $Z$ at each iteration, instead of on $R_i$, we must do it $O(|R_i|)$ time, not $O(|Z|)$ time, in order to retain the linear time bound. To do this, we leave the full/partial/empty labelings for $Z$ from one iteration to the next, so that we only have to update them, using $R_i$, rather than re-creating them each time a new row is considered.

To facilitate this, we keep updated labels $c(X)$ and $n(X)$ on each Venn class $X$, where $c(X)$ denotes the cardinality of $X$ and $n(X)$ is the number of elements of $Z$ in $X$. Labels only need to be updated when a Venn class is split. It is split into $X \cap R_i$ and $X \setminus R_i$. We may find $c(X \cap R_i)$ and $n(X \cap R_i)$ by counting them directly, since there are $O(|R_i|)$ of these elements. The classes are implemented with doubly-linked lists, and these sets are removed from the list for $X$, leaving it to represent $X \setminus R_i$. Subtracting $c(X \cap R_i)$ and $n(X \cap R_i)$ from the old labels $c(X)$ and $n(X)$ gives the updated labels for $c(X \setminus R_i)$ and $n(X \setminus R_i)$ in $O(1)$ time. Each of the new classes is full if its $c()$ and $n()$ labels are equal, empty if its $n()$ label is 0, and partial otherwise.

To evaluate whether one of the conditions of Lemma 5 holds, it is easy to see that it suffices to keep track of *transition pairs*, which are consecutive pairs such that one contains a 0 and one contains a 1. This happens when their full/partial/empty labels are unequal, or else both partial. When a new transition pair forms, we have touched at least one member of the pair within our $O(|R_i|)$ operations, so keeping track of these does not affect this time bound.

Since finding $P$ takes linear time by the BFS of Section 3, it remains only to bound the time required to find the first step, finding the elements $A$ and $B$ of Lemma 7. This is much more straightforward, since we apply it once for each iteration of the loop of Algorithm 1, hence we can afford to take $\Theta(size(M))$ time for the test. We can apply the entire set of full/partial/empty labels for $Z$ to the PQ tree within this bound, by working from the leaves to the root.

For each Q node $Q$, the members of the overlap component whose union is $Q$ are unions of more than one and fewer than all of its children. How to find them in linear time for all Q nodes has been described previously, for example in [9]. We find the rows of the overlap component that contain a Venn child of $Q$ that is labeled full or partial (a "1"). Out of all such rows, let $A'$ be the one with a leftmost right endpoint, and let $B'$ be the one with the rightmost left endpoint. By a simple greedy swapping argument, the overlap component giving rise to $Q$ contains an $A$ and a $B$ satisfying Lemma 7 if and only if $A'$ and $B'$ satisfy it, which happens if and only if there is a child between the right endpoint of $A'$ and the left endpoint of $B'$ that is labeled partial or empty (a "0"). This can also clearly be implemented so that the time bound over all Q nodes takes $O(size(M))$ time.

Once the set $\mathcal{P}_2 \cup \{Z\}$ of rows of a Tucker matrix have been found, it remains to find the columns. This is a set of columns, the removal of any one of which

would undermine the conditions of Lemma 5, which are satisfied initially by $\mathcal{P}_2 \cup \{Z\}$. Deletion of a column undermines the lemma if and only if does one of the following:

1. It disconnects the path in overlap graph on $\mathcal{R}$;
2. It undermines the only remaining 1-0-1 configuration, or 0-1-0 configuration with a 1 in the unconstrained class.

The second test is elementary and omitted because of space constraints. For the first test, recall that $P_2 = (R_1, R_2, \ldots, R_k)$ is a chordless path in the overlap graph. Let $\mathcal{A} = \{R_1 \setminus R_2\} \cup \{X | X = R_i \cap R_{i+1} \text{ for } i \in \{1, 2, \ldots, k-1\}\}$ $\cup \{Y | Y = R_{i+1} \setminus R_i \text{ for } i \in \{1, 2, \ldots, k-1\}\}$. Each element of $\mathcal{A}$ is consecutive-ones ordered, the sum of cardinalities of sets in $\mathcal{A}$ is $O(size(M))$. The overlap graph remains connected if and only if every member of $\mathcal{A}$ contains at least one retained column. We give each column a list of members of $\mathcal{A}$ it is contained in and keep a counter on each element of $\mathcal{A}$ indicating the number of remaining columns it contains. When removing a column $C$, the counters can be updated by decrementing the counters of members of $\mathcal{A}$ in its list. A column cannot be removed if removing it would decrement a counter to 0.

## 5 Finding a Lekkerkerker-Boland Subgraph

Tucker observed that the smallest graphs whose clique matrices contain a Tucker matrix must be exactly the LB graphs [19]. Deletion of rows for the three simplicial vertices in the clique matrix of $G_I$, $G_{II}$, $G_{IV}$ of $G_V$ gives a Tucker matrix. However, it does not follow that every such Tucker submatrix in a clique matrix of $G$ can be extended to a submatrix giving the clique matrix of one of these LB graphs. This is illustrated by Figure 5. Fortunately, it is true for clique matrices of chordal graphs, which we show next.

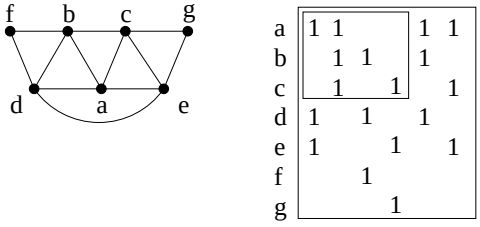


**Fig. 5.** A graph $G$ and its clique matrix. In the upper left of the matrix is a Tucker matrix, $M_{III}$. The only LB subgraph of $G$ is the chordless cycle $(b, c, e, d)$, which does not contain row $a$ of the $M_{III}$. This illustrates that not every Tucker submatrix in a clique matrix can be extended to the clique matrix of an LB subgraph.
.

If $G$ is not chordal, we may return a $G_{III}$ by the algorithm of [18]. Henceforth, we may assume that the graph is chordal. A *clique tree* of a chordal graph is

a tree that has one node for each maximal clique, and with the property that for each vertex $v$ of $G$, the cliques that contain $v$ induce a connected subtree. Every chordal graph has a clique tree, see for example [5]. The following Lemma is immediate from results that appear in [5].

**Lemma 9.** *Let $T$ be a clique tree for a chordal graph $G$, and let $K$ be a leaf. Then $K$ contains a simplicial vertex of $G$. Let $S$ be the simplicial vertices of $K$, and let $T'$ be the result of deleting leaf $K$ from $T$. Deleting $S$ from $G$ yields an induced subgraph that has $T'$ as a clique tree.*

**Definition 3.** *By* shrinking a clique tree *$T$, let us denote the operation of deleting the set $S$ of simplicial vertices in a leaf $K$ of $T$, yielding a smaller graph $G'$ with the smaller clique tree described by Lemma 9.*

**Lemma 10.** *Let $G$ be a chordal graph and let $M_T$ be a submatrix of a clique matrix of $G$ that is an instance of $M_I$ on three vertices or an instance of $M_{II} - M_V$. Then the clique matrix of an LB graph occurs in the submatrix of $K$ induced by the columns of $M_T$, the rows of $M_T$, and three additional rows, one for each of the simplicial vertices depicted in Figure 2.*

*Proof.* Let $T$ be the column numbers occupied by $M_T$ in the clique matrix of $G$, let $i \in T$, let $\mathcal{K}$ be the cliques of $G$ corresponding to columns of $T$, and $K_i$ be the clique of $G$ corresponding to the column $i$. Let $V'$ be vertices corresponding to rows occupied by $M_T$, let $\mathcal{C} = \{K \cap V' | K \in \mathcal{K}\}$ and let $C_i = K_i \cap V'$. Let $\mathcal{C}' = \{C' \setminus C_i | C' \in \mathcal{C} \text{ and } C' \neq C_i\}$. If the intersection graph of $\mathcal{C}'$ is connected, then let us say that $C_i$ is an *outsider* in $M_T$.

If $C_i$ is an outsider, then in any clique tree of $G$, $K_i$ does not lie on the path between any pair of members of $\mathcal{K}$ in the clique tree of $G$. This is seen as follows. Suppose $K_i$ lies on the path $P$ between two members of $\mathcal{K}$. Then removal of $K_i$ from the clique tree separates the clique tree into two or more trees, at least two of which contain members of $\mathcal{K}$. Because the intersection graph of $\mathcal{C}'$ is connected, there exists $v \in V' \setminus C_i$ that resides in cliques in two of these trees. Since $v \notin C_i$, the subtree of the clique tree induced by cliques containing $v$ is not connected, a contradiction.

Therefore, suppose we iteratively shrink the clique tree of $G$ subject to the constraint that we do not shink any member of $\mathcal{K}$ when it becomes a leaf. The procedure halts when all leaves of the clique tree of the resulting graph $G'$ are members of $\mathcal{K}$. If $C_i$ is an outsider, $K_i$ is a leaf in the clique tree of $G'$, which means that it has a simplicial vertex $s$, by Lemma 9. The only column of $T$ where $s$ has a 1 is in column $i$, hence $C_i$ is the set of neighbors of $s$ in $G'$.

It is easily verified that in $M_{II}(k)$, the first and last two columns of $M_{II}(k)$ ($\{1, k-1\}$, $\{0, k-2\}$, $\{k-2, k-1\}$) are outsiders. These are the neighbor sets of $G_V(k+3)$. Therefore, the $M_{II}(k)$ can be extended to a submatrix that is the clique matrix of $G_V(k+3)$. The rows of this submatrix is an induced $G_V(k+3)$ in $G$, as illustrated on the righthand side of Figure 2. Similarly, the first, third and fifth column of $M_{IV}$ and the first, third and fourth column of $M_V$ the first column and last two columns of $M_{III}(k)$, and all columns of $M_I(3)$ are outsiders.

Any instance of these submatrices in the clique matrix of a chordal graph can be extended to an instance of $G_I$, $G_{II}$, $G_{IV}(k+3)$, and $G_V(6)$, respectively, identifying an LB subgraph of $G$.

## References


[1] S. Booth and S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13:335–379, 1976.

[2] A. Brandstaedt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics, Philadelphia, 1999.

[3] C. Chauve, U.-U. Haus, T. Stephen, and V. P. You. Minimal conflicting sets for the consecutive ones property in ancestral genome reconstruction. *Journal of Computational Biology*, 17:1167–1181, 2010.

[4] M. Dom, J. Guo, and R. Niedermeier. Approximation and fixed-parameter algorithms for consecutive ones submatrix problems. *Journal of Computer and System Sciences*, 76:204–221, 2010.

[5] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[6] P. Hell and J. Huang. Certifying LexBFS recognition algorithms for proper inteval graphs and proper interval bigraphs. *SIAM J. Discrete Math*, 18:554–570, 2004.

[7] D. Kratsch, R.M. McConnell, K. Mehlhorn, and J.P. Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. *SIAM Journal on Computing*, 36:326–353, 2006.

[8] C. Lekkerker and D. Boland. Representation of finite graphs by a set of intervals on the real line. *Fund. Math.*, 51:45–64, 1962.

[9] G. S. Lueker and K. S. Booth. A linear time algorithm for deciding interval graph isomorphism. *J. ACM*, 26:183–195, 1979.

[10] R. M. McConnell. A certifying algorithm for the consecutive-ones property. *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA04)*, 15:761–770, 2004.

[11] R. M. McConnell, K. Mehlhorn, S. Näher, and P. Schweitzer. Certifying algorithms. *Computer Science Reviews*, 5:119–161, 2011.

[12] J. Meidanis, O. Porto, and G.P. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88:325–354, 1998.

[13] Fred S. Roberts. *Graph Theory and Its Applications to Problems of Society.* Society for Industrial and Applied Mathematics, Philadelphia, 1978.

[14] D. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5:266–283, 1976.

[15] J. Spinrad. *Efficient Graph Representations.* American Mathematical Society, Providence RI, 2003.

[16] J Stoye and R. Wittler. A unified approach for reconstructing ancience gene clusters. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6:387–400, 2009.

[17] M. Tamayo. *Algorithms for Finding Tucker Patterns.* PhD thesis, Simon Fraser University, 2013.

[18] E. E. Tarjan and M. Yannakakis. Addendum: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 14:254–255, 1985.

[19] A. Tucker. A structure theorem for the consecutive 1's property. *Journal of Combinatorial Theory, Series B*, 12:153–162, 1972.

[20] G. Wegner. *Eigenschaften der Nerven homologishe-einfacher Familien im* $R^n$. PhD thesis, Universität Göttingen, 1967.