Discrete Applied Mathematics 157 (2009) 3721–3727

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A new characterization of matrices with the consecutive ones property

N.S. Narayanaswamy [a,*], R. Subashini [b]

[a] Indian Institute of Technology Madras, Chennai, India

[b] National Institute of Technology, Calicut, India

ARTICLE INFO

Article history:
Received 13 September 2007
Received in revised form 28 February 2009
Accepted 27 August 2009
Available online 17 October 2009

Keywords:
Interval graphs
0–1 matrices
Consecutive ones property

ABSTRACT

We consider the following constraint satisfaction problem: Given a set F of subsets of a finite set S of cardinality n, and an assignment of intervals of the discrete set $\{1, \ldots, n\}$ to each of the subsets, does there exist a bijection $f : S \to \{1, \ldots, n\}$ such that for each element of F, its image under f is same as the interval assigned to it. An interval assignment to a given set of subsets is called *feasible* if there exists such a bijection. In this paper, we characterize feasible interval assignments to a given set of subsets. We then use this result to characterize matrices with the Consecutive Ones Property (COP), and to characterize matrices for which there is a permutation of the rows such that the columns are all sorted in ascending order. We also present a characterization of set systems which have a feasible interval assignment.

1. Introduction

The COP is an interesting and fundamental combinatorial property of binary matrices. The COP appears in many applications; data retrieval, DNA physical mapping, sequence assembly, interval graph recognition, and recognizing Hamiltonian cubic graphs. Testing if a given graph is an interval graph, and testing if a given cubic graph is Hamiltonian are applications of algorithms for testing if a given 0–1 matrix has COP. The maximal clique-vertex incidence matrix is tested for COP to check if a given graph is an interval graph [5]. Similarly, from [12] a cubic graph is Hamiltonian if and only if the matrix $A + I$ has a permutation of rows that leaves at most two blocks of consecutive ones in each column. A is the adjacency matrix of the given graph and I is the identity matrix. Testing if a matrix has COP is also applied for constructing physical maps by hybridization (see [9]), and testing if a database has the consecutive retrieval property (see [4]). To ask for a permutation of the rows such that each column is sorted is a natural extension of the COP. For 0–1 matrices this question is studied as the concept of 1-drop matrices in [2].

Previous work. The first mention of COP, according to D.G. Kendall [8], was made by Petrie, an archaeologist, in 1899. Some heuristics were proposed for testing the COP in [11] before the work of Fulkerson and Gross [3] who presented the first polynomial time algorithm. Subsequently Tucker [13] presented a characterization of matrices with the COP based on certain forbidden matrix configurations. Booth and Lueker [1] proposed the first linear time algorithm for the problem using a powerful data structure called the PQ-Tree. This data structure exists if and only if the given matrix has the COP. Hsu [7] presented another linear time algorithm for testing COP without using PQ-trees. More recently in 2001, he introduced [6] a new data structure called PC tree as a generalization of PQ-Tree. This was used to test if a binary matrix has the CiRcular Ones Property (CROP). Another generalization of the PQ-tree is the PQR-tree introduced by Meidanis and Munuera [10]. This generalization was a nice extension of the approach of Booth and Leuker so that PQR-trees are defined even for matrices that do not possess the COP. Further, for matrices that do not have the COP, the PQR-tree points out specific subcollections of columns responsible for the absence of the COP [9]. In 2003, an almost linear time algorithm has been proposed [9] to construct a PQR-tree.

* Corresponding author. Tel.: +914422574369; fax: +914422574352.
E-mail addresses: swamy@cse.iitm.ernet.in, narayanaswamy@gmail.com (N.S. Narayanaswamy).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2009.08.001

Our work. Our motivation in this work was to understand the Consecutive Ones Testing (COT) algorithm due to [7] and to extend it to finding a permutation of the rows of matrix such that the columns are all sorted. Clearly, to sort just one column, we can easily identify a family of row permutations that achieves the sorting. So for each column in a given matrix we can associate a set of *sorting permutations*. The question now is whether the intersection of these sets, one per column, is empty or not? In this paper we identify a natural succinct representation of the *sorting permutations* of a column. This leads to the question that we pose in the abstract: Given an interval assignment to a set system, is it feasible? We then present a necessary and sufficient condition for an interval assignment to be feasible. In particular, we show that an interval assignment to a set system is feasible if and only if it preserves the cardinality of the intersection of every pair of sets. While a feasible interval assignment must necessarily satisfy this property, to our surprise we do not find this characterization in the literature, definitely not explicitly to the best of our knowledge. We use this characterization to characterize matrices with the COP, and characterize matrices whose columns can be sorted by a row permutation. We also show a necessary and sufficient condition for a feasible interval assignment to exist. Our proofs are all constructive and can be easily converted into algorithms that run in polynomial time in the input size. An important consequence of this work is what we view as the modularization of COT algorithm due to Hsu [7]. Two essential modules in the COT algorithm are to find a feasible interval assignment for the columns of a 0-1 matrix, and then to find a permutation that is witness to the feasibility of the interval assignment. Our study in this paper can also be seen as a different angle of study, and yet along the line of work initiated by Meidanis et al. [10,9]. In their work, they study the set system associated with the columns of the matrix. In particular their results find a *closure* of the set system which also has the COP if the given set system has the COP. In this paper, we take another natural approach to study the set system associated with the columns of the matrix. We consider the set of row permutations that yield consecutive ones in the columns of a matrix. We then ask how this set gets pruned when a new column is added to the matrix. In the process of answering this question, we use the decomposition of the given matrix into prime matrices as done in [7]. Our work also opens up natural generalizations of the COP. For example, given a matrix is there a permutation of the rows such that in each column the rows are partitioned into at most two sorted sets of consecutive rows?. This would be an interesting way to classify matrices, and the combinatorics of this seems very interesting and non-trivial. This would also be a natural combinatorial generalization of the k-drop property for 0-1 matrices which is studied in [2] and references therein.

Roadmap. In Section 2.2 we present a characterization of *feasible* interval assignments, and its consequence to COT. The main part of this section is the algorithm to find a permutation that realizes a given interval assignment. Following this in Section 3 we state our characterization of set systems that have a feasible interval assignment.

2. Characterization of feasible interval assignments

In this paper $\{A_1, \ldots, A_m\}$ is a set of subsets of $\{1, \ldots, n\}$. Let $r_i = |A_i|$, $1 \le i \le m$. An interval assignment to $\{A_1, \ldots, A_m\}$ is the set $\{(A_i, B_i) | 1 \le i \le m, B_i \subseteq \{1, \ldots, n\}$, and elements of B_i are consecutive$\}$. In our presentation, B_i is used to denote the interval assigned to A_i, $1 \le i \le m$. Further, an interval here is a set of consecutive integers from the set $\{1, \ldots, n\}$. An Intersection Cardinality Preserving Interval Assignment (ICPIA) to $\{A_1, \ldots, A_m\}$ is a set of ordered pairs $\{(A_i, B_i) | 1 \le i \le m\}$ such that for each i, $1 \le i \le m$, $|A_i| = |B_i|$, and for every two sets A_i and A_j, $|A_i \cap A_j| = |B_i \cap B_j|$. We also use the ordered pair (P, Q) to denote the assignment of interval Q to the set P. Since in each ordered pair (P, Q), $|P| = |Q|$, we also use (P, Q) to represent all permutations of $\{1, \ldots, n\}$ such that the set P is mapped to the interval Q. An interval assignment $\{(A_i, B_i) | 1 \le i \le m\}$ is defined to be *feasible* if there is a permutation of $\{1, \ldots, n\}$ such that for each $1 \le i \le m$, the image of A_i under the permutation is the interval B_i. Two intervals are said to be *strictly intersecting* if their intersection is non-empty and neither is contained in the other.

Theorem 1. *If an interval assignment $\{(A_i, B_i) | 1 \le i \le m\}$ is feasible, then it is an ICPIA.*

Proof. Since the interval assignment $\{(A_i, B_i) | 1 \le i \le m\}$ is feasible, there is a permutation σ such that $\sigma(A_i) = B_i$, $1 \le i \le m$. Since σ is a permutation it follows that $|A_i| = |B_i|$. Further, for the same reason, for all $1 \le i, j \le m$, $\sigma(A_i \cup A_j) = B_i \cup B_j$, and therefore $|A_i \cap A_j| = |B_i \cap B_j|$. Consequently, the interval assignment is an ICPIA. Hence our claim. □

2.1. Feasible permutations from an ICPIA

We now show that given an ICPIA $\{(A_i, B_i) | 1 \le i \le m\}$, there is a permutation σ of $\{1, \ldots, n\}$ such that $\sigma(A_i) = B_i$, $1 \le i \le m$. Without loss of generality, we assume that the ordered pairs in the ICPIA are indexed according to the order obtained by sorting the left end point of the intervals B_i in the ICPIA, and ties are broken by sorting in ascending order of right end points. In other words, the interval B_1 has the smallest left end point among all intervals and the interval B_m has the largest left end point.

Before we outline the algorithm for constructing a feasible permutation from the ICPIA, we prove the following two crucial lemmas.

Lemma 1. *Let $(A_1, B_1), (A_2, B_2), (A_3, B_3)$ be elements of an ICPIA. Then, $|A_1 \cap A_2 \cap A_3| = |B_1 \cap B_2 \cap B_3|$.*

Proof. If for any two intervals the intersections are empty, then the corresponding sets have empty intersection, and therefore, it follows that the intersection of the 3 intervals is empty, and so is the intersection of the 3 sets. The claim is true in this case. Therefore, we consider the case when the pairwise intersection of the intervals is non-empty. By the Helly Property, if a set of intervals are such that the pairwise intersection is non-empty, then the intersection of all the intervals in the set is also non-empty. Further, it is also clear that if three intervals have a non-empty intersection, then one of the intervals is contained in the union of the other two. Without loss of generality, let $B_3 \subseteq B_1 \cup B_2$, therefore $|B_1 \cup B_2 \cup B_3| = |B_1 \cup B_2| = |A_1 \cup A_2| \leq |A_1 \cup A_2 \cup A_3|$.

We next prove that $|B_1 \cup B_2 \cup B_3| \geq |A_1 \cup A_2 \cup A_3|$. Since B_1, B_2, and B_3 are intervals, it is also clear that $|B_1 \cap B_2 \cap B_3| = \min\{|B_1 \cap B_2|, |B_1 \cap B_3|, |B_2 \cap B_3|\}$. Without loss of generality, let us assume that $|B_1 \cap B_2 \cap B_3| = |B_2 \cap B_3|$. Applying this to the Inclusion–Exclusion formula for $|B_1 \cup B_2 \cup B_3|$, we get $|B_1 \cup B_2 \cup B_3| = |B_1| + |B_2| + |B_3| - |B_1 \cap B_2| - |B_1 \cap B_3|$. The r.h.s is in turn equal to $|A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_1 \cap A_3| \geq |\{A_2 \cup A_3\} \setminus A_1| + |A_1| = |A_1 \cup A_2 \cup A_3|$. Therefore, it follows that $|A_1 \cup A_2 \cup A_3| = |B_1 \cup B_2 \cup B_3|$. From, the given hypothesis and the Inclusion–Exclusion formula it now follows that $|A_1 \cap A_2 \cap A_3| = |B_1 \cap B_2 \cap B_3|$. Hence the proof. □

Corollary 1. *Let $(A_1, B_1), (A_2, B_2), (A_3, B_3)$ be elements of an ICPIA. Then, $|(A_1 \setminus A_2) \cap A_3| = |(B_1 \setminus B_2) \cap B_3|$.*

Proof. Clearly, $|(A_1 \setminus A_2) \cap A_3| = |(A_1 \setminus (A_1 \cap A_2)) \cap A_3| = |(A_1 \cap A_3| - |(A_1 \cap A_2)) \cap A_3|$. From Lemma 1 we know that $|(A_1 \cap A_2) \cap A_3| = |(B_1 \cap B_2) \cap B_3|$, and that $|A_1 \cap A_3| = |B_1 \cap B_3|$ follows from the fact that we have an ICPIA. Therefore, it follows that $|A_1 \cap A_3| - |(A_1 \cap A_2) \cap A_3| = |B_1 \cap B_3| - |(B_1 \cap B_2) \cap B_3| = |(B_1 \setminus B_2) \cap B_3|$. Hence the corollary. □

Algorithm 1 Permutations from an ICPIA $\{(A_i, B_i) | 1 \leq i \leq m\}$

Let $\Pi_0 = \{(A_i, B_i) | 1 \leq i \leq m\}$
$j = 1$;
while There is $(P_1, Q_1), (P_2, Q_2) \in \Pi_{j-1}$ with Q_1 and Q_2 strictly intersecting **do**
  $\Pi_j = \Pi_{j-1} \setminus \{(P_1, Q_1), (P_2, Q_2)\}$;
  $\Pi_j = \Pi_j \cup \{(P_1 \cap P_2, Q_1 \cap Q_2), (P_1 \setminus P_2, Q_1 \setminus Q_2), (P_2 \setminus P_1, Q_2 \setminus Q_1)\}$;
  j = j+1;
end while
$\Pi = \Pi_j$;
Return Π;

We now prove a set of invariants which are used to show that for each *j*, in the *j*th iteration of Algorithm 1, Π_j represents the set $\{\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\} | \sigma$ is a permutation, and $\sigma(P) = Q$, for each (P,Q) $\in \Pi_j\}$.

Lemma 2. *At the end of the jth iteration, $j \geq 0$, of the while loop of Algorithm 1, the following three are invariant*

- *Invariant* I*: Q is an interval for each $(P, Q) \in \Pi_j$.*
- *Invariant* II*: $|P| = |Q|$ for each $(P, Q) \in \Pi_j$.*
- *Invariant* III*: For any two $(P', Q'), (P'', Q'') \in \Pi_j$, $|P' \cap P''| = |Q' \cap Q''|$.*

Proof. The proof of the lemma is by induction on *j*, which is the number of times the while loop has executed. For $j = 0$, by definition, $\Pi_0 = \{(A_i, B_i) | 1 \leq i \leq m\}$. All the invariants hold because we are dealing with an ICPIA. Therefore the base case is proved. Let us assume that the lemma holds for $j - 1$. We now show that the lemma holds for *j*. First, invariant I holds due to the following reason: If $(P, Q) \in \Pi_j$ and Π_{j-1}, then by the induction hypothesis Q is an interval, $|P| = |Q|$, and invariant II also holds. If $(P, Q) \in \Pi_j$, but not in Π_{j-1}, then it means that (P, Q) is one of the following three pairs for some $(P_1, Q_1), (P_2, Q_2) \in \Pi_{j-1}$ such that Q_1 and Q_2 are strictly intersecting: $(P_1 \cap P_2, Q_1 \cap Q_2)$, or $(P_1 \setminus P_2, Q_1 \setminus Q_2)$, or $(P_2 \setminus P_1, Q_2 \setminus Q_1)$. By invariant III of the induction hypothesis, it follows that $|P| = |Q|$. Since the Q_1 and Q_2 are strictly intersecting, it follows that Q is an interval. To prove invariant III, let us consider a pair $(P', Q'), (P'', Q'') \in \Pi_j$. If both are in Π_{j-1}, then invariant III holds. If one of them is not in Π_{j-1}, then it is one of the following three pairs for some $(P_1, Q_1), (P_2, Q_2) \in \Pi_{j-1}$ where Q_1 and Q_2 are strictly intersecting: $(P_1 \cap P_2, Q_1 \cap Q_2)$, or $(P_1 \setminus P_2, Q_1 \setminus Q_2)$, or $(P_2 \setminus P_1, Q_2 \setminus Q_1)$. Now applying Lemma 1 and Corollary 1, it follows that in this case too for each pair $(P', Q'), (P'', Q'') \in \Pi_j$, $|P' \cap P''| = |Q' \cap Q''|$. Therefore the induction hypothesis is proved. Hence the lemma. □

Theorem 2. *Let $\{(A_i, B_i) | 1 \leq i \leq m\}$ be an ICPIA. Then, there is a permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $\sigma(A_i) = B_i$.*

Proof. To prove this theorem we use the output of Algorithm 1 and *refine* it further to a set $\{(A'_i, B'_i) | 1 \leq i \leq m\}$ such that the permutations represented are consistent with the permutations represented by the output, and further all the A'_is (B'_is) are pairwise disjoint, so clearly the permutation claimed the theorem exists. Consider Π output by Algorithm 1 for $\{(A_i, B_i) | 1 \leq i \leq m\}$. For the sake of ease, we add (A_0, B_0) to Π, where $A_0 = B_0 = \{1, \ldots, n\}$. Clearly, from the algorithm, for any two $(P_1, Q_1), (P_2, Q_2) \in \Pi$, either Q_1 and Q_2 are disjoint, or one is contained in the other. In other words, they cannot be strictly intersecting. So to further refine Π, we consider the following tree, which can be called a containment tree. The nodes of this tree represent $(P, Q) \in \Pi$. Let (P_1, Q_1) and (P_2, Q_2) be the elements of Π associated with two nodes. There is an edge from the node corresponding to (P_1, Q_1) to the node corresponding to (P_2, Q_2) if and only if Q_1 is the smallest interval that contains Q_2, among all the ordered pairs in Π. The root of the tree is the pair (A_0, B_0). Since the Q_is are intervals, this data structure is a tree which we denote by T. We now refine Π as outlined in Algorithm 2 using the function call Post-Order-Traversal $(T, (A_0, B_0), \Pi)$. Let the resulting set be Π_{end} which is a set of ordered pairs (P_i, Q_i), $1 \leq i \leq m'$ where $m' \geq m$ is a finite number. In an ordered pair $(P_i, Q_i) \in \Pi_{end}$, Q_i is not necessarily an interval. However, for any two $(P_1, Q_1), (P_2, Q_2) \in \Pi_{end}$, $|P_1 \cap P_2| = |Q_1 \cap Q_2| = 0$, and $|P_i| = |Q_i|$. The other property is that for j such that $1 \leq j \leq m$ the image of A_j remains B_j. The reason is that each (A_j, B_j) is only *broken* into smaller sets in both Algorithm 1 and Algorithm 2. Therefore, any permutation that maps P_i to Q_i for each $(P_i, Q_i) \in \Pi_{end}$ satisfies all the constraints specified by the ICPIA $\{(A_i, B_i)|1 \leq i \leq m\}$. Hence, Π_{end} represents a family of permutations such that for each permutation σ, $\sigma(A_i) = B_i$, $1 \leq i \leq m$. □

Algorithm 2 Permutations from Π obtained from Algorithm 1

```
function Post-Order-Traversal(T,root-node,Π)
if (root-node is a leaf) then
  return
end if
while (root-node has unexplored children) do
  next-root-node = an-unexplored-child-of-root-node
  Post-Order-Traversal(T,next-root-node,Π)
end while
if (root-node has no unexplored children) then
  Let (P, Q) denote the element of Π associated with root-node
  Let (P1, Q1) ... (Pk, Qk) be the pairs associated with the children of root-node
  Π ← Π \ {(P, Q)}
  Π ← Π ∪ {(P \ (P1 ∪ ... ∪ Pk), Q \ (Q1 ∪ ... ∪ Qk))}
  return
end if
```

Theorems 1 and 2 together prove that an interval assignment $\{(A_i, B_i)|1 \leq i \leq m\}$ is feasible if and only if it is an ICPIA. We now use this result to characterize matrices whose rows can be rearranged to obtain desired *interval-properties* on the columns. The basic idea is to associate a set system with each column based on the desired property, and then test if the resulting problem instance has an ICPIA.

2.2. Characterizing matrices with the COP

Definition and Notation: An $m \times n$ matrix M with 0–1 entries is said to have the consecutive ones property (COP) if there is a permutation of the rows such that in the resulting matrix the ones occur consecutively in each column. Such a permutation is said to leave consecutive ones in the columns. Our characterization of matrices with the COP provides a new analysis of a recent Consecutive Ones Testing algorithm [7]. For each $1 \leq i \leq m$, let $A_i = \{p|M_{pi} = 1\}$ Let $r_i = |A_i|$ denote the number of ones in the ith column. Let $B_i = \{l, l+1, \ldots, l+r_i-1\}$ denote an interval assigned to the ith column, where $1 \leq l \leq n$. The following theorem holds as an application of the results obtained in the previous section in a more general setting.

Theorem 3. *A 0–1 matrix M has the COP if and only if there exists an ICPIA* $\{(A_i, B_i)|1 \leq i \leq m\}$.

The problem of finding a permutation of the rows of a matrix such that each column is sorted in ascending order can be solved by creating a natural interval assignment the same lines as outlined for testing the COP. For sorting the columns the interval assignment is straightforward: in each column, the index of each row containing a 0 must be mapped to a number smaller than the image of the index of any row containing a 1.

3. Structural characterization of matrices with an ICPIA

In this section we address the question of whether a given set system $\{A_i \subseteq \{1, \ldots, n\}|1 \leq i \leq m\}$ has an ICPIA. Quite naturally we view the given set system as an $n \times m$ binary matrix M. In M, the jth column corresponds to the set A_j and $M_{ij} = 1$ if and only if $i \in A_j$, otherwise $M_{ij} = 0$. Note that the columns of M are distinct. In the rest of this section, we say that a matrix M has an ICPIA if there is an ICPIA $\{(A_i, B_i)|1 \leq i \leq m\}$ where $A_i = \{p|M_{pi} = 1\}$. We also refer to the jth column of M as the set A_j. The word *set* is used to refer to the set associated with a column, and any other meaning is explicitly clarified. We recall the notion of matrix decomposition introduced in [7].

An undirected graph on the columns of M: With the given matrix M, associate an undirected graph $G(M)$ where the vertices correspond to A_i, $1 \leq i \leq m$. We assume that vertex v_i corresponds to set A_i. $\{v_i, v_j\} \in E(G(M))$ if and only if the corresponding sets intersect and neither is contained in the other. A prime sub-matrix of M is a matrix formed by a set of columns of M which correspond to a connected component of the graph G. Let us denote the prime sub-matrices by $M_1, \ldots, M_p$. Clearly, two distinct matrices have a distinct set of columns. Let $col(M_i)$ be the set of columns in the sub-matrix

M_i. We also introduce the notation for the support of a prime sub-matrix M_i; $\text{supp}(M_i) = \bigcup_{j \in col(M_i)} A_j$. Note that for each i, $\text{supp}(M_i) \subseteq \{1, \ldots, n\}$. For a set of prime sub-matrices X we define $\text{supp}(X) = \bigcup_{M \in X} \text{supp}(M)$.

A partial order on the prime sub-matrices: Consider the relation $\preccurlyeq$ on the prime sub-matrices $M_1, \ldots, M_p$ defined as follows:

$$\{(M_i, M_j) |\ \text{A set } S \in M_i \text{ is contained in a set } S' \in M_j\} \cup \{(M_i, M_i) | 1 \leq i \leq p\}. \tag{1}$$

Lemma 3. *Let $(M_i, M_j) \in \preccurlyeq$. Then there is a set $S' \in M_j$ such that for each $S \in M_i$, $S \subseteq S'$.*

Proof. Since $(M_i, M_j) \in \preccurlyeq$, it follows, by definition of $\preccurlyeq$, that there is an $S' \in M_j$ and $S \in M_i$ such that $S \subseteq S'$. We want to prove that each set of M_i is contained in S'. We prove this by contradiction. Let $T \in M_i$ be the first vertex in a path in $G(M_i)$ from $S \in M_i$ such that $T \not\subseteq S'$. Let $T' \in M_i$ be the neighbor of T on the path. Clearly, $T' \subseteq S'$. Further $T \cap T' \neq \phi$, and neither is contained in the other. Therefore, $T \cap S' \neq \phi$. By our assumption, $T \not\subseteq S'$. Therefore, $T \in M_j$. This is a contradiction to the fact that two distinct prime sub-matrices have distinct sets of columns. Therefore, our assumption of the existence of T is wrong. Hence the lemma. □

Lemma 4. *For each pair of prime sub-matrices, either $(M_i, M_j) \notin \preccurlyeq$ or $(M_j, M_i) \notin \preccurlyeq$.*

Proof. The proof is by contradiction. If we assume that for two distinct i and j, $(M_i, M_j) \in \preccurlyeq$ and $(M_j, M_i) \in \preccurlyeq$, then from Lemma 3 that there is an $S \in M_i$ such that each $S' \in M_i$ is contained in S. Since the columns of M are distinct, this is a contradiction to the definition of M_i. Therefore, our assumption is wrong. Hence the lemma is proved. □

Lemma 5. *If $(M_i, M_j) \in \preccurlyeq$ and $(M_j, M_k) \in \preccurlyeq$, then $(M_i, M_k) \in \preccurlyeq$.*

Proof. This follows from Lemma 3 and the definition of containment. □

Lemma 6. *If $(M_i, M_j) \in \preccurlyeq$ and $(M_i, M_k) \in \preccurlyeq$, then either $(M_j, M_k) \in \preccurlyeq$ or $(M_k, M_j) \in \preccurlyeq$.*

Proof. The proof is by contradiction. Let us assume that both (M_j, M_k) and (M_k, M_j) are not in $\preccurlyeq$. Along with the fact that M_j and M_k are prime sub-matrices, this implies that $\text{supp}(M_j)$ and $\text{supp}(M_k)$ are disjoint. Further, from Lemma 3 we know that $\text{supp}(M_i)$ is strictly contained in $\text{supp}(M_j)$ and $\text{supp}(M_k)$. This is a contradiction to the conclusion that $\text{supp}(M_j) \cap \text{supp}(M_k) = \phi$ which follows from the assumption that (M_j, M_k) and (M_k, M_j) are not in $\preccurlyeq$. Hence the lemma. □

Theorem 4. *$\preccurlyeq$ is a partial order on the set of prime sub-matrices of M. Further, it uniquely partitions the prime sub-matrices of M such that on each set in the partition $\preccurlyeq$ induces a total order.*

Proof. This follows from the previous four lemmas and the fact that $\preccurlyeq$ is reflexive by definition. □

Lemma 7. *A 0–1 matrix M has an ICPIA if and only if each prime sub-matrix has an ICPIA.*

Proof. If M has an ICPIA, then by definition each prime sub-matrix has an ICPIA. We now prove the reverse direction by construction. Let us assume that each M_i, $1 \leq i \leq p$ has an ICPIA. Let $X_1, \ldots, X_l$ be the partition mentioned in Theorem 4. From the definition of a prime sub-matrix and the definition of $\preccurlyeq$ it follows that $\text{supp}(X_r) \cap \text{supp}(X_s) = \phi$ for each $1 \leq r \neq s \leq l$. Therefore, to complete our construction, we identify an interval $I(X_k)$, $1 \leq k \leq l$, and then prove our claim for a generic set in the partition. The interval $I(X_k)$ is written as $[l(X_k), r(X_k)]$. Here $l(X_1) = 1$, $r(X_k) = l(X_k) + |\text{supp}(X_k)| - 1$, for $1 \leq k \leq l$, and $l(X_k) = r(X_{k-1}) + 1$ for $2 \leq k \leq l$. Clearly, $I(X_k)$ is the interval which will contain the intervals assigned to the columns in the matrix formed by the prime sub-matrices in X_k. We next prove the claim for a generic set, say X_k, in the partition. Let $X_k = \{M_{1k}, M_{2k}, \ldots, M_{j_k k}\}$ with $M_{j_k k} \preccurlyeq \ldots \preccurlyeq M_{2k} \preccurlyeq M_{1k}$. From the definition of $\preccurlyeq$, for each r, $2 \leq r \leq j_k$, $\text{supp}(M_{rk})$ is contained in at least one set in $M_{(r-1)k}$. Therefore, it follows that $\text{supp}(X_k) = \text{supp}(M_{1k})$. For the construction, we associate an interval with each prime sub-matrix in X_k. For $j_k \geq r \geq 2$, Let C_{rk} denote the set of intervals assigned to those sets of $M_{(r-1)k}$ which contain $\text{supp}(M_{rk})$. We define $I(M_{rk}) = \bigcap_{I \in C_{rk}} I$. The interval associated with M_{1k} is $I(M_{1k}) = [l(X_k), l(X_k) + |\text{supp}(M_{1k})| - 1]$. For $1 \leq r \leq j_k$, let us consider the interval I' obtained by taking the union of intervals in an ICPIA associated with M_{rk}; we have this by the hypothesis. We know that $|I'| = |\text{supp}(M_{rk})|$ since I' is the set of intervals obtained from an ICPIA assigned to the sets in M_{rk}. Further, for each r, $1 \leq r \leq j_k$, $|\text{supp}(M_{rk})| \leq |I(M_{rk})|$. Therefore, $|I'| \leq |I(M_{rk})|$. To complete the construction, we order the elements of I' from the smallest point to the largest point, and map the ith rank element of I' to the ith rank element of $I(M_{rk})$. Clearly, this bijection takes each interval in the ICPIA given by the hypothesis and yields an ICPIA that is completely contained in $I(M_{rk})$. This construction yields an ICPIA for the prime sub-matrices of X_k such that each interval in this assignment is contained in $I(X_k)$. Consequently, this yields an ICPIA for M. Hence the reverse direction is proved, and consequently the lemma is proved. □

3.1. An algorithm for finding an ICPIA

Here we show that it is possible to find an ICPIA to the columns of a given binary matrix M in polynomial time, provided there is one. Algorithm 3 is based on the structural characterization described above in this section and algorithm 4. In algorithm 3 the function *ICPIA*$(M', I(M'))$ assigns an ICPIA to a prime sub-matrix M' in the interval $I(M') = [l(M'), r(M')]$. Basically, the function *ICPIA*$(M', I(M'))$ is a loop that calls Algorithm 4 for each column of M'.

Algorithm 3 Algorithm to find an ICPIA for a matrix M

Identify the prime sub-matrices. This is done by constructing the strict overlap graph and identify connected components. Each connected component yields a prime sub-matrix.
Construct the partial order $\preccurlyeq$ on the set of prime sub-matrices.
Construct the partition $X_1, \dots, X_l$ of the prime sub-matrices induced by $\preccurlyeq$ and find $I(X_k)$.
Construct the total order on each set in the partition.
for $(k = 1; k \leq l; k++)$ **do**
  Let $I(M_{1k}) = [l(X_k), l(X_k) + supp(M_{1k}) - 1]$
  $ICPIA(M_{1k}, I(M_{1k}))$
  for $(r = 2; r \leq j_k; r++)$ **do**
    Construct C_{rk} from the ICPIA assigned to sets of $M_{(r-1)k}$.
    Let $I(M_{rk}) = \bigcap_{I \in C_{rk}} I$
    $ICPIA(M_{rk}, I(M_{rk}))$
  end for
end for

In algorithm 4, the elements of the set $\{S^1, \dots, S^p\}$ are the sets corresponding to p columns, of M', that have been assigned an ICPIA among them. Let this ICPIA be $\{I^1, \dots, I^p\}$. Further, let S^1 be the set such that the sets of M' that intersect with it have a pairwise non-empty intersection. The interval I^1 assigned to S^1 is $[l(M'), l(M') + |S^1| - 1]$. Now, let S denote the set corresponding to the jth column such that S has a non-empty intersection with some S^i, $1 \leq i \leq p$, and $S \not\subseteq S^i$, $S^i \not\subseteq S$. Algorithm 4 describes how S is assigned an interval I such that $\{I^1, \dots, I^p, I\}$ is an ICPIA for $\{S^1, \dots, S^p, S\}$.

Algorithm 4 Basic step in an algorithm to find an ICPIA for a prime matrix M'

ICPIA(Set S, Integer $p > 0$)
/* $S \cap S^i \neq \phi$ for some $i \in \{1, \dots, p\}$, but $S \not\subseteq S^i$, $S^i \not\subseteq S$.
Assigns to S an interval I such that $\{I^1, \dots, I^p, I\}$ forms an ICPIA for $\{S^1, \dots, S^p, S\}$.
*/
  Let $|S \cap S^i| = z$.
  Let I_l be the interval such that $|I_l \cap I^i| = z$, $|I_l| = |S|$ and the z common elements are the smallest elements of I^i.
  Let I_r be the interval such that $|I_r \cap I^i| = z$, $|I_r| = |S|$, and the z common elements are the largest elements of I^i.
  if $p == 1$ **then**
    Assign I_l to S
    /* In this case, I_r could also be assigned to S. This will yield the **other** ICPIA */
  else
    if $|I_l \cap I^q| = |S \cap S^q|$ for each $q \in \{1, \dots, p\}$ **then**
      Assign I_l to S and exit.
    end if
    if $|I_r \cap I^q| = |S \cap S^q|$ for each $q \in \{1, \dots, p\}$ **then**
      Assign I_r to S and exit.
    end if
  end if
  Report no ICPIA and exit.

Theorem 5. *Algorithm 4 outputs an ICPIA to a prime matrix M' iff there is an ICPIA for M'.*

Proof. The only-if part of the theorem is straightforward. We now show that if there is an ICPIA for M', then Algorithm 4 will indeed discover it. The key fact is that in M' for each set S, there is another set $T \in M'$ such that $S \cap T \neq \phi$, and S and T are not contained in each other. Due to this fact, there are exactly two ICPIAs for M'. The two distinct ICPIAs differ based on the interval assigned to S_1, see Algorithm 4. If I_l is assigned to S_1, then we get one, and the other ICPIA is obtained by assigning I_r to S_1. For each subsequent set, say S^j, the interval to be assigned is forced. It is forced due to the fact that the interval assigned to S^j is based on the interval assigned to S^i, where $S^i \cap S^j \neq \phi$, and $S^i \not\subseteq S^j$, and $S^j \not\subseteq S^i$. Given the fact that the algorithm is an exact implementation of these observations, it follows that Algorithm 4 finds an ICPIA if there is one. □

4. Conclusion

We have introduced the notion of an ICPIA formally and have shown that an interval assignment is feasible if and only if it is an ICPIA. We then use this observation to characterize matrices that have the consecutive ones property, thus giving a newer understanding of Hsu's algorithm [7] for COT. This combinatorial understanding also leads to a characterization of matrices whose rows can be permuted so that each column is sorted. Finally, we have also presented an algorithm to test if a set system has an ICPIA using approaches developed by [7].

Acknowledgements

We gratefully acknowledge the comments of the anonymous referees which have increased the readability of this paper.

References

[1] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, Journal of Computer System Science (1976).
[2] R. Chandrasekaran, S.N. Kabadi, S. Lakshminarayanan, An extension of a theorem of Fulkerson and Gross, Linear Algebra and its Applications 246 (1996) 23–29.
[3] D. Fulkerson, O.A Gross, Incidence matrices and interval graphs, Pacific Journal of Mathematics (1965).
[4] S. Ghosh, File organization: The consecutive retrieval property, Communications of the ACM (1979).
[5] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, 1980.
[6] W.L. Hsu, PC-trees vs. PQ-trees, in: Proc. of the 7th Annual International Conference on Computing and Combinatorics, in: Lecture Notes in Computer Science, vol. 2108, 2001, pp. 207–217.
[7] W.L. Hsu, A simple test for the consecutive ones property, Journal of Algorithms 42 (2002) 1–16.
[8] D. Kendall, Incidence matrices, interval graphs and seriation in archaeology, Pacific Journal of Mathematics (1969).
[9] O.P.J. Meidanis, G. Telles, On the consecutive ones property, Discrete Applied Mathematics 88 (1998) 325–354.
[10] J. Meidanis, E. Munuera, A theory for the consecutive ones property, in: Proceedings of the III South American Workshop on String Processing, August 8–9, 1996, pp. 194–202.
[11] W.S. Robinson, A method for chronologically ordering archaelogical deposits, American Antiquity (1951).
[12] R. Wang, F.C.M. Lau, Y.C. Zhao, Hamiltonicity of regular graphs and blocks of consecutive ones in symmetric matrices, Discrete Applied Mathematics 155 (17) (2007) 2312–2320.
[13] A. Tucker, A structure theorem for the consecutive ones property, Journal of Combinatorial Theory, Series B 12 (2) (1972) 153–162.