Generalization of the Consecutive-ones Property

A THESIS

submitted by

ANJU SRINIVASAN

for the award of the degree of

MASTER OF SCIENCE *by Research*

from the department of

COMPUTER SCIENCE AND ENGINEERING

at

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Guindy, Chennai - 600036

JANUARY 2012

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **i**

ABSTRACT **iii**

LIST OF TABLES **vi**

LIST OF FIGURES **vii**

ABBREVIATIONS **viii**

NOTATION **ix**

1 Introduction **1**

1.1 Organization of the document 1

1.2 Illustration of the problem 1

1.3 Basic preliminaries - general definitions and nomenclature . . . 5

1.4 Consecutive-ones Property Testing - a Brief Survey 5

1.4.1 Matrices with COP 5

1.4.2 Optimization problems in COP 7

1.5 ********* Application of COP in Areas of Graph Theory and Algorithms 9

1.5.1 ********* COP in Relational Database Model 9

1.5.2 ********* COP in Graph Isomorphism 9

1.5.3 ********* Certifying Algorithms 9

1.6 Generalization of COP - the Motivation 9

1.7 Summary of New Results in this Thesis 11

2 Consecutive-ones Property – A Survey of Important Results **14**

2.1 COP in Graph Theory 14

2.2 Matrices with COP 16

2.3 Optimization problems in COP 16

2.4 ********* COP in Relational Database Model 16

2.5 ********* COP in Graph Isomorphism 17

2.6 ********* Certifying Algorithms 17

2.6.1 Matrices with COP 17

2.6.2 Optimization problems in COP 19

3 Tree Path Labeling of Path Hypergraphs - the New Results 20

3.1 Introduction 20

3.2 Preliminaries to new results 22

3.3 Characterization of Feasible Tree Path Labeling 26

3.4 Computing feasible TPL with special target trees [c1] 37

[c1] give problem definition etc

3.4.1 Target tree is a Path 38

3.4.2 Target tree is a k-subdivided Star 38

3.5 TPL with no restrictions 42

3.5.1 Finding an assignment of tree paths to a set system . . . 43

3.6 Complexity of Tree Path Assignment-A Discussion 48

3.6.1 Consecutive Ones Testing is in Logspace 48

4 Conclusion 51

REFERENCES [c2] 53

[c2] *minor*: make names in bib file uniform style w.r.t. firstname/initials

LIST OF TABLES

1.1 Students and study groups in *Wallace Studies Institute* 2

1.2 A solution to study group accomodation problem 2

2.1 **** Graph matrices 15

LIST OF FIGURES

1.1 *Infinite Loop* street map. 4

1.2 *Infinite Loop* street map with study group routes allocated. 4

1.3 Solution to the student accommodation problem. 4

1.4 Examples of k-subdivided stars. (a) $k = 0$ (b) $k = 2$ 12

2.1 Matrices defined in Def. 2.1.1 15

2.2 Matrices with and without COP. 17

3.1 (a) 8-subdivided star with 7 rays (b) 3-subdivided star with 3 rays 38

CHAPTER 2

Consecutive-ones Property – A Survey of Important Results

This chapter surveys several results that are significant to this thesis or to COP in general. These predominantly pertain to characterizations of COP, algorithmic tests to check for COP (COT), optimization problems on binary matrices that do not have COP and some applications of COP.

c1 [c1] *important*: AL have a few lines about organization of chapter

c2 [c2] DEFINE COP SOMEWHERE

2.1 COP in Graph Theory

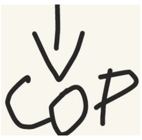
COP is closely connected to several types of graphs by way of describing certain combinatorial graph properties. There are also certain graphs, like convex bipartite graphs, that are defined solely by some of its associated matrix having COP. In this section we will see the relevance of consecutive-ones property to graphs. To see this we introduce certain binary matrices that are used to define graphs in different ways. While adjacency matrix is perhaps the most commonly used such matrix, Definition 2.1.1 defines this and a few more.

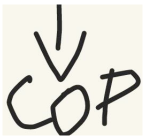
Definition 2.1.1
Matrices that define graphs.[Dom08, Def. 2.4] Let G and H be defined as follows. $G = (V, E_G)$ is a graph with vertex set $V = \{v_i \mid i \in [n]\}$ and edge set $E_G \subseteq \{(v_i, v_j) \mid i, j \in [n]\}$ such that $|E_G| = m$. $H = (A, B, E_H)$ is a bipartite graph with partitions $A = \{a_i \mid i \in [n_a]\}$ and $B = \{b_i \mid i \in [n_b]\}$.

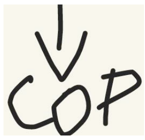
2.1.1–i. *Adjacency matrix* of G is the symmetric $n \times n$ binary matrix M with $m_{i,j} = \mathbf{1}$ if and only if $(v_i, v_j) \in E_G$ for all $i, j \in [n]$.

2.1.1–ii. *Augmented adjacency matrix* of G is obtained from its adjacency matrix by setting all main diagonal elements to $\mathbf{1}$, i. e. $m_{i,i} = \mathbf{1}$ for all $i \in [n]$.

2.1.1–iii. *Maximal clique matrix* or *vertex-clique incidence matrix* of G is the $n \times k$ binary matrix M with $m_{i,j} = \mathbf{1}$ if and only if $v_i \in C_j$ for all $i \in [n], j \in [k]$ where $\{C_j \mid j \in [k]\}$ is the set of maximal cliques of G.

$\mathbf{G_1}$: 

$\mathbf{G_2}$: 

$\mathbf{H}$: 

C	w_1	w_2	w_3	w_4
u_1	**1**	**1**	0	0
u_2	**1**	**1**	**1**	**1**
u_3	0	0	**1**	**1**
u_4	0	**1**	**1**	0
u_5	**1**	**1**	**1**	0

$\mathbf{A_1}$	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	**1**	**1**	0	**1**	**1**
v_2	**1**	0	**1**	**1**	0	**1**
v_3	**1**	**1**	0	**1**	**1**	0
v_4	0	**1**	**1**	0	0	0
v_5	**1**	0	**1**	0	0	0
v_6	**1**	**1**	0	0	0	0

B	c_1	c_2	c_3	c_4
v_1	**1**	**1**	**1**	0
v_2	**1**	0	0	0
v_3	0	**1**	0	0
v_4	0	0	**1**	**1**
v_5	0	**1**	**1**	0
v_6	0	0	0	**1**

$\mathbf{A_2}$	v_1	v_2	v_3	v_4	v_5	v_6
v_1	**1**	**1**	**1**	0	**1**	**1**
v_2	**1**	**1**	**1**	**1**	0	**1**
v_3	**1**	**1**	**1**	**1**	**1**	0
v_4	0	**1**	**1**	**1**	0	0
v_5	**1**	0	**1**	0	**1**	0
v_6	**1**	**1**	0	0	0	**1**

$\mathbf{A'_2}$	v_1	v_6	v_2	v_4	v_3	v_5
v_1	**1**	**1**	**1**	0	**1**	**1**
v_6	**1**	**1**	**1**	0	0	0
v_2	**1**	**1**	**1**	**1**	**1**	0
v_4	0	0	**1**	**1**	**1**	0
v_3	**1**	0	**1**	**1**	**1**	**1**
v_5	**1**	0	0	0	**1**	**1**

Figure 2.1: A_1 is the *adjacency matrix* and A_2 is the *augmented adjacency matrix* of G_1. Matrix A'_2 is obtained from A_2 by permuting its rows and columns to achieve *CROP order*, i. e. A_2 has CROP – thus G_1 is *a concave-round graph* (Def. 2.1.2 ii) and *a circular-arc graph* (Tab. 2.1) B is the maximal clique matrix of G_2 and has COP – thus G_2 is an *interval graph*(Def. 2.1.2 ii). C is the half adjacency matrix of bipartite graph H and has COP on rows – thus H is a convex bipartite graph.

2.1.1–iv. *Half adjacency matrix* of H is the $n_a \times n_b$ binary matrix M with $m_{i,j} = \mathbf{1}$ if and only if $(a_i, b_j) \in E_H$

□

Now we will see in Definition 2.1.2 certain graph classes that is related to COP or CROP.

`tab 2.1 –` c1

c1 have an abridged version of table 2.1 in dom

Table 2.1: **** Graph matrices

Definition 2.1.2

Graphs that relate to COP.[Dom08, Def. 2.5] Let G be a graph and H be a bipartite graph.

2.1.2–i. G is *convex-round* if its adjacency matrix has the CROP.

2.1.2–ii. G is *concave-round* if its augmented adjacency matrix has CROP. [c2] [c3]

c2 cite BHY00

c3 *minor:* add CROP to glossary

2.1.2–iii. G is an *interval graph* if its vertices can be mapped to intervals on the real line such that two vertices are adjacent if and only if their corresponding intervals overlap [c1]. G is an interval graph if and only if its maximal clique matrix has COP due to the result by [GH64] which states that the maximal cliques of interval graph G can be linearly ordered such that for all $v \in V(G)$, cliques containing v are consecutive in the ordering.

c1 cite Ben59, Haj57

2.1.2–iv. G is a *unit interval graph* if it is an interval graph such that all intervals have the same length.

2.1.2–v. G is a *proper interval graph* if it is an interval graph such that no interval properly contains another.

2.1.2–vi. G is a *circular-arc graph* if its vertices can be mapped to a set of arcs on a circle such that two vertices are adjacent if and only if their corresponding arcs overlap.

2.1.2–vii. H is *convex bipartite* if its half adjacency matrix has COP on either rows or columns.

2.1.2–viii. H is *biconvex bipartite* or *doubly convex*[YC95] if its half adjacency matrix has COP on both rows and columns.

2.1.2–ix. H is *circular convex* if its half adjacency matrix has CROP.

□

c2 c3 c4c5c6 c7c8c9c10c11c12

c2 *pressing:* ADD As it will be described in detail later in this document, isomorphism of certain

c3 *pressing:* ADD: peo exists iff chordal. lexicographic BFS [tag:chordalGraph]

c4 *pressing:* ADD: A well known result in

c5 *pressing:* verify from paper the statement of claim.

c6 *pressing:* max clique vertex incidence matrix of

c7 *pressing:* citation?!!

c8 *pressing:* cite uses these results to give the first polynomial time algorithm for COT.

c9 *pressing:* check. how do they use it?

c10 *pressing:* A bipartite graph is convex

c11 *pressing:*

2.2 Matrices with COP

c13

Figure 2.2 shows examples of consecutive-ones property. [c14]

2.3 Optimization problems in COP

c15

2.4 ********* COP in Relational Database Model

c16 c17

M_1:				M_1':				M_2:			
c_1	c_2	c_3	c_4	c_3	c_1	c_4	c_2	d_1	d_2	d_3	d_4
1	0	**1**	0	**1**	**1**	0	0	**1**	**1**	0	0
0	**1**	0	**1**	0	0	**1**	**1**	0	**1**	**1**	0
1	0	0	**1**	0	**1**	**1**	0	0	**1**	0	**1**

Figure 2.2: Matrices with and without COP. M_1 has COP because by permuting its columns, c_1-c_4, one can obtain M_1' where the **1**s in each row are consecutive. M_2, however, does not have COP since no permutation of its columns, d_1-d_4, will arrange **1**s in each row consecutively [Dom08].

2.5 ********* COP in Graph Isomorphism

c18 c19

c18 Expand on sec:appgraphiso

c19 (canonization theme)

2.6 ********* Certifying Algorithms

c1 c2

c1 Expand on sec:appcertalgo

c2 (certification McC04 theme)

– REFERENCE CONTENT –

2.6.1 Matrices with COP

As seen earlier, the interval assignment problem (illustrated as the course scheduling problem in Section 1.2), is a special case of the problem we address in this thesis, namely the tree path labeling problem (illustrated as the study group accomodation problem). The interval assignment problem and COP problem are equivalent problems. In this section we will see some of the results that exists in the literature today towards solving the COP problem and optimization problems surrounding it.

Recall that a matrix with COP is one whose rows (columns) can be rearranged so that the **1**s in every column (row) are in consecutive rows (columns). COP in binary matrices has several practical applications in diverse fields including scheduling [HL06], information retrieval [Kou77] and computational biology [ABH98]. Further, it is a tool in graph theory [Gol04] for interval graph recognition, characterization of Hamiltonian graphs, planarity testing [BL76] and in integer linear programming [HT02, HL06].

The obvious first questions after being introduced to the consecutive ones property of binary matrices are if COP can be detected efficiently in a binary matrix and if so, can the COP permutation of the matrix also be computed efficiently? Recognition of COP in a binary matrix is polynomial time solvable and the first such algorithm was given by [FG65]. A landmark result came a few years later when [Tuc72] discovered the families of forbidden submatrices that prevent a matrix from having COP and most, if not all, results that came later were based on this discovery which connected COP in binary matrices to convex bipartite graphs. In fact, the forbidden submatrices came as a corollary to the discovery that convex bipartite graphs are AT-free in [Tuc72][c3]. The first linear time algorithm for COP testing (COT) was invented by [BL76] using a data structure called PQ trees. Since then several COT algorithms have been invented – some of which involved variations of PQ trees [MM96, Hsu01, McC04], some involved set theory and ICPIA [Hsu02, NS09], parallel COT algorithms[AS95, BS03, CY91] and certifying algorithms[McC04].

c3 check

The construction of PQ trees in [BL76] draws on the close relationship of matrices with COP to interval graphs. A PQ tree of a matrix is one that stores all row (column) permutations of the matrix that give the COP orders (there could be multiple orders of rows or columns) of the matrix. This is constructed using an elaborate linear time procedure and is also a test for planarity[c1]. PQR trees is a generalized data structure based on PQ trees [MM96, MPT98]. [TM05] describes an improved algorithm to build PQR trees. [c2][Hsu02] describes the simpler algorithm for COT. Hsu also invented PC trees [Hsu01][c3] which is claimed to be much easier to implement. [NS09] describes a characterization of consecutive-ones property solely based on the cardinality properties of the set representations of the columns (rows); every column (row) is equivalent to a set that has the row (column) indices of the rows (columns) that have one entries in this column (row). This is interesting and relevant, especially to this thesis because it simplifies COT to a great degree. [c4]

c1 check check check. both interval graph and planarity in this paper?

c2 improv in terms of what?

c3 This result first appeared inproc ISAAC92

c4 it reduces the solution search space. fill in the blanks.

[McC04] describes a different approach to COT. While all previous COT algorithms gave the COP order if the matrix has the property but exited stating negative if otherwise, this algorithm gives an evidence by way of a certificate of matrix even when it has no COP. This enables a user to verify the algorithm's result even when the answer is negative. This is significant from an implementation perspective because automated program verification is hard and manual verification is more viable. Hence having a certificate reinforces an implementation's credibility. Note that when the matrix *has* COP, the COP order is the certificate. The internal machinery of this algorithm is related to the weighted betweenness problem addressed[c5] in [COR98]. [c6] [c7]

c5 in what way??

c6 expand on the COP order graph creation and it having to be bipartite for M to have COP. and thus an odd cycle being an evidence of no COP.

c7 where should this

2.6.2 Optimization problems in COP

So far we have been concerned about matrices that have the consecutive ones property. However in real life applications, it is rare that data sets represented by binary matrices have COP, primarily due to the noisy nature of data available. At the same time, COP is not arbitrary and is a desirable property in practical data representation [COR98, JKC+04, Kou77]. In this context, there are several interesting problems when a matrix does not have COP but is "close" to having COP or is allowed to be altered to have COP. These are the optimization problems related to a matrix which does not have COP. Some of the significant problems are surveyed in this section.

[c1c2] [Tuc72] showed that a matrix that does not have COP have certain substructures that prevent it from having COP. Tucker classified these forbidden substructures into five classes of submatrices. This result is presented in the context of convex bipartite graphs which [Tuc72] proved to be AT-free[c3]. By definition, convex bipartite graph have half adjacency matrices that have COP on either rows or columns (graph is biconvex if it has COP on both)[Dom08]. A half adjacency matrix is a binary matrix representing a bipartite graph as follows. The set of rows and the set of columns form the two partitions of the graph. Each row node is adjacent to those nodes that represent the columns that have **1**s in the corresponding row. [Tuc72] proves that this bipartite graph has no asteroidal triple if and only if the matrix has COP and goes on to identify the forbidden substructures for these bipartite graphs. The matrices corresponding to these substructures are the forbidden submatrices.

c1 – sect 4.1 in cite:d08phd has many results surveyed. hardness results, approx. results. results are usually for a class of matrices (a, b) where number **1**s in columns and rows are restriced to a and b . – problem of flipping at most k entries of M to make it attain COP. this is NP complete cite:b75-phd

c2 (1) scite:lb62 showed that interval graphs are AT-free. describe AT (2) show the close relationship b/w COP and graphs sec 2.2, pg 31

c3 check this up. give details. - doms'

Once a matrix has been detected to not have COP (using any of the COT algorithms mentioned earlier), it is naturally of interest to find out the smallest forbidden substructure (in terms of number of rows and/or columns and/or number of entries that are **1**s). [Dom08] discusses a couple of algorithms which are efficient if the number of **1**s in a row is small. This is of significance in the case of sparse matrices where this number is much lesser than the number of columns. $(*, \Delta)$-*matrices* are matrices with no restriction on number of **1**s in any column but has at most Δ **1**s in any row. Min COS-R (Min COS-C), Max COS-R (Max COS-C) are similar problems which deals with inducing COP on a matrix. In Min COS-R (Min COS-C) the question is to find the minimum number of rows (columns) that must be deleted to result in a matrix with COP. In the dual problem Max COS-R (Max COS-C) the search is for the maximum number of rows (columns) that induces a submatrix with COP. Given a matrix M with no COP, [Boo75] shows that finding a submatrix M' with all columns[c4] but a

c4 check if b75 deals with COP col or COP row. also is it any submatrix with k less than r rows or submatrix must have all columns?

maximum cardinality subset of rows such that M' has COP is NP complete. [HG02] corrects an error of the abridged proof of this reduction as given in [GJ79]. [Dom08] discusses all these problems in detail giving an extensive survey of the previously existing results which are almost exhaustively all approximation results and hardness results. Taking this further, [Dom08] presents new results in the area of parameterized algorithms for this problem[c5].

Another problem is to find the minimum number of entries in the matrix that can be toggled to result in a matrix with COP. [Vel85] discusses approximation of COP Augmentation which is the problem of changing of the minimum number of zero entries to **1**s so that the resulting matrix has COP. As mentioned earlier, this problem is known to be NP complete due to [Boo75]. [Vel85] also proves, using a reduction to the longest path problem, [c1] that finding a Tucker's forbidden submatrix of at least k rows is NP complete. [c2] [c3]

[JKC+04] discusses the use of matrices with almost-COP (instead of one block of consecutive **1**s, they have x blocks, or *runs*, of consecutive **1**s and x is not too large) in the storage of very large databases. The problem is that of reordering of a binary matrix such that the resulting matrix has at most k runs of **1**s. This is proved to be NP hard using a reduction from the Hamiltonian path problem.[c4]
c5c6 c7 c8

c5 elaborate - what are the results?

c1 or is it a survey of another result? check.

c2 how is this different from booth's 75 result??

c3 where should this go? cite—tz04 (approx submatrix with COP sparse matrices)

c4 Theorem 2.1 in jkckv

c5 (1) A connection of COP problem to the travelling salesman problem is also introduced. what does this mean? – COP can be used as a tool to reorder $0.5T \leq runs(M) \leq$ (2) The optimization version of the k-run problem, i.e. minimization of number of blocks of ones is proven to be NP complete by cite:k77

c6 are these two the same?

c7 what is the reduction?

c8 other problems similar to COP – cite:ckl96 (ILP, circ ones, one drop) – cite:th98 (generalization of COP - minimax, biotonic column) Tucker

c9 remove if none.

Chapter Notes

[1]The notion of *feasibility* is formally defined in Section 3.2.

[2]The terms *tree path labeling* and *tree path assignment* are, in informal language, synonyms. Formally, the former refers to the bijection $\ell : \mathcal{F} \to \mathcal{P}$. The latter refers to the set of ordered pairs $\{(S, P) \mid S \in \mathcal{F}, P \in \mathcal{P}\}$. $\mathcal{P}$ is a set of paths on T.

[3]A *hypergraph* is an alternate representation of a set system and will be used in this thesis.

[4]See Section 3.2 for the formal definition.

[5]A *tree path labeling* ℓ is a bijection of paths from the target tree T to the hyperedges in given hypergraph $\mathcal{F}$.

[6]See Section 3.3 for the definition of ICPPL.

[7]The path from a leaf to the root, the vertex with highest degree, is called a *ray* of the k-subdivided star.

[8]The vertex with maximum degree in a k-subdivided star is called *root*.

[9]If there exists an FTPL for a hypergraph $\mathcal{F}$, it is called a path hypergraph.

c9

REFERENCES

[ABH98] J. E. Atkins, E. G. Boman, and B. Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SICOMP: SIAM Journal on Computing*, 28, 1998.

[AS95] Annexstein and Swaminathan. On testing consecutive-ones property in parallel. In *SPAA: Annual ACM Symposium on Parallel Algorithms and Architectures*, 1995.

[BL76] Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using *PQ*-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, December 1976.

[BLS99] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[Boo75] Kellogg S. Booth. *PQ-tree algorithms.* PhD thesis, Univ. California, Berkeley, 1975.

[BP92] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. Technical report, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, 1992.

[BS03] David A. Bader and Sukanya Sreshta. A new parallel algorithm for planarity testing, April 11 2003.

[COR98] Thomas Christof, Marcus Oswald, and Gerhard Reinelt. Consecutive ones and a betweenness problem in computational biology. *Lecture Notes in Computer Science*, 1412, 1998.

[CY91] Lin Chen and Yaacov Yesha. Parallel recognition of the consecutive ones property with applications. *J. Algorithms*, 12(3):375–392, 1991.

[Dom08] Michael Dom. *Recognition, Generation, and Application of Binary Matrices with the Consecutive-Ones Property.* PhD thesis, Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany, 2008. Published by Cuvillier, 2009.

[FG65] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pac. J. Math.*, 15:835–855, 1965.

[Gav78] Fanica Gavril. A recognition algorithm for the intersection graphs of paths in trees. *Discrete Mathematics*, 23(3):211 – 227, 1978.

[GH64] P. C. Gilmore and A. J. Hoffman. A characterization of comparability graphs and of interval graphs. *Can. J. Math.*, 16:539–548, 1964.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability.* Freeman, San Francisco, 1979.

[Gol04] Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57 of *Annals of Discrete Mathematics.* Elsevier Science B.V., 2004. Second Edition.

[HG02] Hajiaghayi and Ganjali. A note on the consecutive ones submatrix problem. *IPL: Information Processing Letters*, 83, 2002.

[HL06] Dorit S. Hochbaum and Asaf Levin. Cyclical scheduling and multi-shift scheduling: Complexity and approximation algorithms. *Discrete Optimization*, 3(4):327–340, 2006.

[Hsu01] Wen-Lian Hsu. PC-trees vs. PQ-trees. *Lecture Notes in Computer Science*, 2108:207–217, 2001.

[Hsu02] Wen-Lian Hsu. A simple test for the consecutive ones property. *J. Algorithms*, 43(1):1–16, 2002.

[HT02] Hochbaum and Tucker. Minimax problems with bitonic matrices. *NETWORKS: Networks: An International Journal*, 40, 2002.

[JKC+04] Johnson, Krishnan, Chhugani, Kumar, and Venkatasubramanian. Compressing large boolean matrices using reordering techniques. In *VLDB: International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers, 2004.

[KKLV10] Johannes Köbler, Sebastian Kuhnert, Bastian Laubner, and Oleg Verbitsky. Interval graphs: Canonical representation in logspace. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:43, 2010.

[KM02] P. S. Kumar and C. E. Veni Madhavan. Clique tree generalization and new subclasses of chordal graphs. *Discrete Applied Mathematics*, 117:109–131, 2002.

[Kou77] Lawrence T. Kou. Polynomial complete consecutive information retrieval problems. *SIAM Journal on Computing*, 6(1):67–75, March 1977.

[Lin92] Steven Lindell. A logspace algorithm for tree canonization (extended abstract). In *STOC*, pages 400–404. ACM, 1992.

[McC04] Ross M. McConnell. A certifying algorithm for the consecutive-ones property. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 2004.

[MM96] J. Meidanis and Erasmo G. Munuera. A theory for the consecutive ones property. In *Proceedings of WSP'96 - Third South American Workshop on String Processing*, pages 194–202, 1996.

[MPT98] Meidanis, Porto, and Telles. On the consecutive ones property. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 88, 1998.

[NS09] N. S. Narayanaswamy and R. Subashini. A new characterization of matrices with the consecutive ones property. *Discrete Applied Mathematics*, 157(18):3721–3727, 2009.

[PPY94] Barry W. Peyton, Alex Pothen, and Xiaoqing Yuan. A clique tree algorithm for partitioning a chordal graph into transitive subgraphs. Technical report, Old Dominion University, Norfolk, VA, USA, 1994.

[Rei84] John H. Reif. Symmetric complementation. *JACM: Journal of the ACM*, 31(2):401–421, 1984.

[Ren70] Peter L. Renz. Intersection representations of graphs by arcs. *Pacific J. Math.*, 34(2):501–510, 1970.

[Sch93] Alejandro A. Schaffer. A faster algorithm to recognize undirected path graphs. *Discrete Applied Mathematics*, 43:261–295, 1993.

[TM05] Guilherme P. Telles and João Meidanis. Building PQR trees in almost-linear time. *Electronic Notes in Discrete Mathematics*, 19:33–39, 2005.

[Tuc72] Alan Tucker. A structure theorem for the consecutive 1's property. *J. Comb. Theory Series B*, 12:153–162, 1972.

[Vel85] Marinus Veldhorst. Approximation of the consecutive ones matrix augmentation problem. *SIAM Journal on Computing*, 14(3):709–729, August 1985.

[YC95] Yu and Chen. Efficient parallel algorithms for doubly convex-bipartite graphs. *TCS: Theoretical Computer Science*, 147:249–265, 1995.