Originally published in *Journal of Computer and System Sciences.* Elsevier B. V. To appear.

Approximation and Fixed-Parameter Algorithms for Consecutive Ones Submatrix Problems☆

Michael Dom, Jiong Guo[1], Rolf Niedermeier

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, 07743 Jena, Germany.
`{michael.dom,jiong.guo,rolf.niedermeier}@uni-jena.de`

Abstract

We develop an algorithmically useful refinement of a forbidden submatrix characterization of 0/1-matrices fulfilling the Consecutive Ones Property (C1P). This characterization finds applications in new polynomial-time approximation algorithms and fixed-parameter tractability results for the NP-hard problem to delete a minimum number of rows or columns from a 0/1-matrix such that the remaining submatrix has the C1P.

Key words: consecutive ones property, circular ones property, forbidden submatrix characterization, NP-hard problem, fixed-parameter tractability, exact algorithms

---

1. Introduction

A 0/1-matrix has the *Consecutive Ones Property (C1P)* if there is a *permutation* of its columns, that is, a finite series of column swappings, that places the 1s consecutive in every row[2]. The C1P of matrices has a long history and it plays an important role in combinatorial optimization, including application fields such as scheduling [5, 23, 24, 46], information retrieval [30], railway optimization [33, 34, 39], or computational biology [1, 2, 3, 7, 38] (see also [9] for a recent survey). It is well-known that it can be decided in linear time whether a given 0/1-matrix has the C1P, and, if so, also a corresponding permutation can be found in linear time [6, 17, 20, 25, 26, 29, 32, 35].[3]

☆Preliminary versions of parts of this paper appeared in the proceedings of the 4th Annual Conference on Theory and Applications of Models of Computation (TAMC '07), held in Shanghai, China, May 2007 [10], and in the proceedings of the 3rd Algorithms and Complexity in Durham (ACiD '07) Workshop, held in Durham, UK, September 2007 [11].

[1] Supported by the Deutsche Forschungsgemeinschaft (DFG), Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

[2] The C1P can be defined symmetrically for columns; we focus on rows here.

[3] The certifying algorithm of McConnell [32] decides whether a given 0/1-matrix has the C1P or not. If it does not have the C1P, then the algorithm generates a "certificate", that is,

The C1P being a desirable property that often leads to efficient algorithms, the natural problem arises what to do if a given matrix does not have the C1P. As a consequence, there has been recently increased interest in matrix modification problems that deal with the transformation of a given 0/1-matrix into a 0/1-matrix fulfilling the C1P [22, 42]. The following three minimization problems show up naturally in this context:

- Find a minimum-cardinality set of *columns* to delete such that the resulting matrix has the C1P. This problem is referred to as MIN-COS-C ("Consecutive Ones Submatrix by Column Deletions").
- Find a minimum-cardinality set of *rows* to delete such that the resulting matrix has the C1P. This problem is referred to as MIN-COS-R ("Consecutive Ones Submatrix by Row Deletions").
- Find a minimum-cardinality set of *1-entries* in the matrix that shall be *flipped* (that is, replaced by 0-entries) such that the resulting matrix has the C1P. This problem is referred to as MIN-CO-1E ("Consecutive Ones by Flipping 1-Entries").

Unfortunately, even for sparse matrices with few 1-entries these quickly turn into NP-hard problems [22, 42]. In this paper, we further explore the algorithmic complexity of these problems, providing new algorithmic results. To this end, based on a "forbidden submatrix characterization" for the C1P due to Tucker [44], our main technical result is a structural theorem dealing with the selection of particularly useful forbidden submatrices. Before we describe our results in more detail, we introduce some notation.

We call a matrix that results from deleting some rows and columns from a given matrix M a *submatrix* of M. Whereas an $m \times n$-matrix is a matrix having m rows and n columns, the term (x, y)*-matrix* will be used to denote a matrix that has at most x 1s per column and at most y 1s per row. (This notation was used in previous work [22, 42].) With $x = *$ or $y = *$, we indicate that there is no upper bound on the number of 1s in columns or in rows, respectively.

Previous work [21, 22, 42] considered the "dual versions" MAX-COS-C and MAX-COS-R of the problems MIN-COS-C and MIN-COS-R. These maximization variants ask for a submatrix M' of a given matrix M such that M' has the C1P and the number d' of the columns (rows) of M' is maximized. The NP-hardness of MAX-COS-C was already mentioned by Garey and Johnson [18], however, Hajiaghayi and Ganjali [21, 22] observed that in Garey and Johnson's monograph [18] the reference for the NP-hardness proof of MAX-COS-C is not correct—indeed, the referenced proof shows the NP-hardness of MAX-COS-R on $(3, 2)$-matrices. Then, MAX-COS-C has been shown NP-hard for $(2, 4)$-matrices by Hajiaghayi and Ganjali [22]. Tan and Zhang showed that for $(2, 3)$- or $(3, 2)$-matrices MAX-COS-C remains NP-hard [42]. Moreover, it

---

a small (compared to the size of the input matrix) proof that can be verified by a "fast and uncomplicated" polynomial-time algorithm (for more details about such certificates see [31]).

turned out that there exists no polynomial-time constant-factor approximation algorithm for Max-COS-C on $(*, 2)$-matrices unless P=NP [42]. The reduction given by Tan and Zhang [42] also shows that Max-COS-C on $(*, 2)$-matrices is W[1]-hard, that is, presumably fixed-parameter intractable, with respect to the parameter "d' = number of columns of M'".[4] On the positive side, Tan and Zhang [42] provided polynomial-time approximability results for the sparsest NP-hard cases of Max-COS-C, that is, for $(2, 3)$- and $(3, 2)$-matrices: Restricted to $(3, 2)$-matrices, Max-COS-C can be approximated within a factor of 0.5; for $(2, *)$-matrices, it is approximable within a factor of 0.5; for $(2, 3)$-matrices, the approximation factor is 0.8. Concerning the minimization versions of the problems, we are only aware of fixed-parameter algorithms and problem kernels for the graph problems 2-Layer Planarization [13, 14, 40, 41] and Linear Arrangement by Deleting Edges [15], which are equivalent to Min-COS-C on $(2, *)$-matrices without identical columns and to Min-COS-R on $(*, 2)$-matrices without identical rows, respectively. Finally, hardness results have been achieved for the related problem of obtaining the C1P by flipping 0-entries [18, 47], and a polynomial-time algorithm is known for the problem of obtaining the C1P by flipping arbitrary entries in a matrix with a constant number of rows or columns [37].

While we use d' to denote the number of columns or rows of the desired submatrix M' when considering the maximization problems Max-COS-C and Max-COS-R, let d denote the number of columns or rows to be deleted from the matrix M to get the submatrix M' having the C1P in the case of the minimization problems Min-COS-C and Min-COS-R. Besides the above mentioned structural theorem, we show the following main algorithmic results.

1. For any constant $\Delta \geq 2$, Min-COS-C on $(*, \Delta)$-matrices is polynomial-time approximable with a factor of 6 if $\Delta = 3$ and with a factor of $(\Delta + 2)$ if $\Delta \neq 3$, and Min-COS-R on $(*, \Delta)$-matrices is polynomial-time approximable with a factor of $(\Delta + 1)$. In particular, this implies a polynomial-time factor-4 approximation algorithm for Min-COS-C on $(*, 2)$-matrices. Factor 4 seems to be the best one can currently hope for because a factor-δ approximation for Min-COS-C restricted to $(*, 2)$-matrices implies a factor-$\delta/2$ approximation for Vertex Cover [42]. It is commonly conjectured that Vertex Cover is not polynomial-time approximable within a factor of $2 - \epsilon$, for any constant $\epsilon > 0$, unless P=NP [27]. Moreover, on $(*, \Delta)$-matrices with $\Delta \geq 2$, Min-COS-C and Min-COS-R are fixed-parameter tractable with respect to the combined parameter Δ, d.

2. On $(*, 2)$-matrices, Min-COS-C and Min-COS-R admit polynomial-time computable problem kernels consisting of $O(d^2)$ columns and rows.

3. On $(2, *)$-matrices, there are polynomial-time approximation algorithms yielding approximation factors of 6 and 4 for Min-COS-C and Min-COS-R,

---

[4] This was independently observed in our previous conference paper [10], see also the PhD thesis of the first author [8].

respectively. Moreover, MIN-COS-C and MIN-COS-R can be solved in $O(6^d \cdot \min\{m^4n, m^2n^3\})$ and $O(4^d \cdot \min\{m^4n, m^2n^3\})$ time, respectively.

We summarize known and new results for MAX-COS-C, MIN-COS-C, MAX-COS-R, and MIN-COS-R in Table 1.

The paper is structured as follows. After a section with preliminaries and basic facts, we consider $(*, \Delta)$-matrices in Sections 3–7: The main idea of our algorithms for these matrices is presented in Section 3. Section 4 contains the proof for our main structural theorem, Sections 5 and 6 deal with two subproblems that need to be considered: finding forbidden submatrices and handling matrices that are already well-structured in some sense. The running times and approximation factors of our algorithms for $(*, \Delta)$-matrices are provided in Section 7. Section 8 briefly describes a kernelization in the case of $(*, 2)$-matrices and approximation and fixed-parameter algorithms for $(2, *)$-matrices. Some open problems are stated in Section 9.

2. Preliminaries and Basic Facts

Given an instance of a minimization (or maximization) problem, a *factor-δ approximation algorithm* for this problem returns in polynomial time a solution such that if the cost of the solution is d and the cost of an optimal solution is d_{opt}, then $d \leq \delta \cdot d_{\text{opt}}$ (or $d \geq \delta \cdot d_{\text{opt}}$, respectively). For an overview on approximation algorithms, refer to [4, 45].

Parameterized complexity is a two-dimensional framework for studying the computational complexity of problems [12, 16, 36]. One dimension is the input size n (as in classical complexity theory), and the other one is the *parameter d* (usually a positive integer). A problem is called *fixed-parameter tractable* (FPT) if it can solved in $f(d) \cdot n^{O(1)}$ time, where f is a computable function only depending on d. A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction rules*, often yielding a *reduction to a problem kernel (kernelization)*. Here the goal is, given any problem instance x together with parameter d, to transform it into a new instance x' with parameter d' such that the size of x' is bounded from above by some function only depending on d, the instance (x, d) has a solution iff (x', d') has a solution, and $d' \leq d$ (see [19] for a recent survey).

By $\mathbb{N}$ we refer to the set of positive integers. For an integer n, let

$$\text{pred}_n,\ \text{succ}_n : \{1, \ldots, n\} \to \{1, \ldots, n\}$$

be the two functions given by

$$\text{pred}_n(x) = \begin{cases} x - 1 & \text{if } x > 1 \\ n & \text{if } x = 1 \end{cases} \qquad \text{and} \qquad \text{succ}_n(x) = \begin{cases} x + 1 & \text{if } x < n \\ 1 & \text{if } x = n. \end{cases}$$

All graphs in this work are undirected. Given a graph $G = (V, E)$ and a vertex $v \in V$, we write $N(v)$ to denote the set of v's neighbors in G, and $N[v]$ to denote the closed neighborhood of v, that is, $N[v] = N(v) \cup \{v\}$. For $V' \subseteq V$,

Table 1: Summary of known and new results for Max-COS-C, Min-COS-C, Max-COS-R and Min-COS-R. The table shows the factors of the approximation algorithms and the running times of the fixed-parameter algorithms. The type (x, y) of the input matrix describes the maximum number of 1s per row and column: An (x, y)-matrix has at most x 1s per column and at most y 1s per row. With $x = *$ or $y = *$, we indicate that there is no upper bound on the number of 1s in columns or in rows, respectively; Δ stands for an any number between 1 and n. We only emphasize the exponential parts of the running times, that is, the shown running times have to be multiplied with polynomials with respect to the input size. An empty field means that we are not aware of any results concerning the corresponding problem variant.

Type	Max-COS-C	Min-COS-C	Max-COS-R	Min-COS-R
$(3, 2)$	• 0.5-approx.[1]	Pos. results: see $(*, 2)$	Pos. results: see $(*, 2)$	Pos. results: see $(*, 2)$
$(*, 2)$	• No const. approx.[2] • W[1]-hard[2,3]	• No 2.72-approx.[2] • Poly. kernel[4] More pos. results: $(*, \Delta)$	• 0.75-approx.[5] • $2^{O(d')}$-alg[5]	• Poly. kernel[4,6] More pos. results: $(*, \Delta)$
$(*, \Delta)$	Neg. results: see $(*, 2)$	• $(\Delta+2)$-approx.[7] • $(\Delta+2)^d \cdot \Delta^{O(\Delta)}$-alg.[7] Neg. results: see $(*, 2)$		• $(\Delta+1)$-approx. • $(\Delta+1)^d \cdot (2\Delta)^{2d}$-alg.
$(2, 3)$	• 0.8-approx.[1] More pos. results: $(2, *)$	Pos. results: see $(2, *)$		Pos. results: see $(2, *)$
$(2, *)$	• 0.5-approx.[1] • $2^{O(d')}$-alg.[5]	• 6-approx. • 6^d-alg.[8]	• No const. approx.[5] • W[1]-hard[3,5]	• No 2.72-approx.[5] • 4-approx. • 4^d-alg.
$(\Delta, *)$			Neg. results: see $(2, *)$	Neg. results: see $(2, *)$

---

[1] Result is due to Tan and Zhang [42].

[2] The hardness of approximating Max-COS-C was shown independently by Dom et al. [10] and Tan and Zhang [42]; the W[1]-hardness of Max-COS-C and the hardness of approximating Min-COS-C with a factor better than 2.72 follow from both reductions [10, 42].

[3] W[1]-hardness is with respect to the parameter d'.

[4] The polynomial problem kernel is with respect to the parameter d.

[5] Result is described in the first author's PhD thesis [8].

[6] More results are known for the case where the $(*, 2)$-matrix does not have duplicate rows: the problem is then equivalent to Linear Arrangement By Deleting Edges, for which a time-$(2.4676^d \cdot |M|^{O(1)})$ algorithm and a smaller problem kernel exist [15].

[7] For the ease of presentation, at this point the table ignores the case $\Delta = 3$. Indeed, if $\Delta = 3$, then the factor of the approximation algorithm for Min-COS-C is 6, and the running time of the fixed-parameter algorithm is $6^d \cdot \Delta^{O(\Delta)} \cdot |M|^{O(1)}$.

[8] More results are known for the case where the $(2, *)$-matrix does not have duplicate columns: the problem is then equivalent to 2-Layer Planarization, for which faster running times [14, 40, 41] and a problem kernel [13] are known.

$G[V']$ denotes the subgraph of G induced by the vertices from V', that is, the graph with vertex set V' and edge set $\{\{u,v\} \in E \mid u,v \in V'\}$. A *hole* is an induced cycle of length at least 5, that is, a cycle of length at least 5 such that there is no edge between two vertices that are not consecutive on the cycle.

We only consider 0/1-matrices $M = (m_{i,j})$, that is, matrices containing only 0s and 1s. We use the term *line* of a matrix M to denote a row or column of M. A column of M that contains only 0-entries is called a 0-*column.* Two matrices M and M' are called *isomorphic* if M' is a permutation of the rows and columns of M. *Complementing* a line ℓ of a matrix means that all 1-entries of ℓ are replaced by 0s and all 0-entries are replaced by 1s. One can regard a matrix as a set of columns together with an order on this set; this order is called the *column ordering* of the matrix.

Let $M = (m_{i,j})$ be a matrix. Let r_i denote the i-th row and let c_j denote the j-th column of M, and let M' be the submatrix of M that results from deleting all rows except for $r_{i_1},\ldots,r_{i_p}$ and all columns except for $c_{j_1},\ldots,c_{j_q}$ from M. Then M'*contains an entry* $m_{i,j}$ of M, denoted by $m_{i,j} \in M'$, if $i \in \{i_1,\ldots,i_p\}$ and $j \in \{j_1,\ldots,j_q\}$. A row r_i of M *belongs* to M', denoted by $r_i \in M'$, if $i \in \{i_1,\ldots,i_p\}$. Analogously, a column c_j of M belongs to M' if $j \in \{j_1,\ldots,j_q\}$. A matrix M is said to *contain a matrix* M' if M' is isomorphic to a submatrix of M.

Every 0/1-matrix $M = (m_{i,j})$ can be interpreted as the adjacency matrix of a bipartite graph G_M: For every line of M there is a vertex in G_M, and for every 1-entry $m_{i,j}$ in M there is an edge in G_M connecting the vertices corresponding to the i-th row and the j-th column of M. We call G_M the *representing graph* of M. In the following definitions, all terms are defined in analogy to the corresponding terms in graph theory: Two lines ℓ,ℓ' of M are *connected in* M if there is a path in G_M connecting the vertices corresponding to ℓ and ℓ'. A submatrix M' of M is called *connected* if each pair of lines belonging to M' is connected in M'. A maximal connected submatrix of M is called a *component* of M. A *shortest path* between two connected submatrices M_1, M_2 of M is the shortest sequence $\ell_1,\ldots,\ell_p$ of lines such that $\ell_1 \in M_1$ and $\ell_p \in M_2$ and the vertices corresponding to $\ell_1,\ldots,\ell_p$ form a path in G_M. If such a shortest path exists, then $p-1$ is called the *distance* between M_1 and M_2.

Note that each submatrix M' of M one-to-one corresponds to an induced subgraph of G_M and that each component of M one-to-one corresponds to a connected component of G_M. An illustration of the components of a matrix is shown in Fig. 1. If the distance between two lines ℓ_1 and ℓ_p is a positive even number, then ℓ_1 and ℓ_p are either both rows or both columns; if the distance is odd, then exactly one of ℓ_1 and ℓ_p is a row and one is a column.

Observation 2.1. *Let M be a matrix and let ℓ be a line of M. Then ℓ belongs to exactly one component M' of M and M' contains all 1-entries of ℓ.*

The following corollary is a direct consequence of Observation 2.1.

Corollary 2.1. *Let M be a matrix and let $M_1,\ldots,M_i$ be the components of M. If the column (or row) sets $F_1,\ldots,F_i$ are optimal solutions for* Min-COS-C

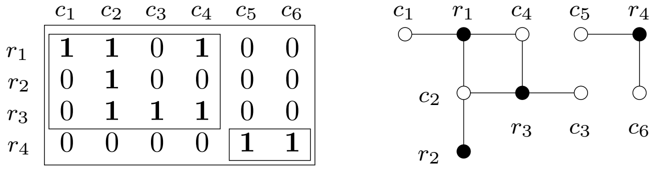

Figure 1: A matrix with two components and its representing bipartite graph.

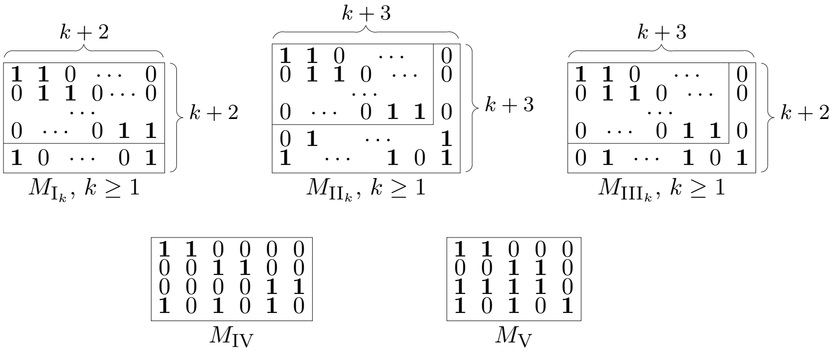

Figure 2: The set T of forbidden submatrices due to Tucker [44] mentioned in Theorem 2.1.

(or Min-COS-R*) on* $M_1, \ldots, M_i$*, respectively, then* $F_1 \cup \ldots \cup F_i$ *is an optimal solution for* Min-COS-C *(or* Min-COS-R*) on* M*.*

Tucker [44] showed that matrices that have the C1P can be characterized by a set of *forbidden submatrices.* This result forms the base of most of our findings.

Theorem 2.1 ([44, Theorem 9])**.** *A matrix* M *has the C1P iff it contains none of the matrices* M_{I_k}*,* M_{II_k}*,* M_{III_k} *(with* $k \geq 1$*),* M_{IV}*, and* M_{V} *(see Fig. 2).*

We denote the set of submatrices given by Theorem 2.1 with T.

The Circular Ones Property, which is defined as follows, is closely related to the C1P, but is easier to achieve. We use it as an intermediate concept for dealing with the harder to achieve C1P.

Definition 2.1. *A matrix has the* Circular Ones Property (Circ1P) *if there exists a permutation of its columns such that in each row of the resulting matrix the 1s appear consecutively or the 0s appear consecutively (or both).*

Intuitively, if a matrix has the Circ1P, then there is a column permutation such that the 1s in each row appear consecutively when the matrix is wrapped around a vertical cylinder. We have no theorem similar to Theorem 2.1 that characterizes matrices having the Circ1P; the following theorem of Tucker [43] is helpful instead.

Theorem 2.2 ([43, Theorem 1]). *Form the matrix M' from a matrix M by complementing all rows with a 1 in the first column of M. Then M has the Circ1P iff M' has the C1P.*

In our context, the following direct consequence of Theorem 2.2 is particularly useful.

Corollary 2.2. *Let M be an $m \times n$-matrix and let j be an arbitrary integer with $1 \leq j \leq n$. Form the matrix M' from M by complementing all rows with a 1 in the j-th column of M. Then M has the Circ1P iff M' has the C1P.*

We end with two straightforward observations. First, note that with respect to (2,2)-matrices, all problems (Min-COS-C, Min-COS-R, and Min-CO-1E) are polynomial-time solvable. The reason is that any $(*, 2)$-matrix can be interpreted as a graph, and, hence, Min-COS-C, Min-COS-R, and Min-CO-1E can be formulated as graph modification problems (see [8, 22] and Section 8). These graph modification problems are polynomial-time solvable on input graphs with maximum degree 2, which correspond to $(2, 2)$-matrices. Second, on $(*, 2)$-matrices the problems Min-COS-R and Min-CO-1E are equivalent because deleting a row one-to-one corresponds to flipping a 1-entry since a row with only one 1-entry can be clearly omitted from further consideration.

3. Outline of the Algorithmic Framework for $(*, \Delta)$-Matrices

In what follows, we briefly describe the basic algorithmic approach underlying all our algorithms. Based on this algorithmic skeleton, we will point out in the subsequent sections the essential ideas needed for deriving our algorithms.

In order to derive constant-factor polynomial-time approximation algorithms or fixed-parameter algorithms for Min-COS-C and Min-COS-R on $(*, \Delta)$-matrices, we exploit Theorem 2.1 by iteratively searching and destroying in the given input matrix every submatrix that is isomorphic to one of the forbidden submatrices from the set T given in Theorem 2.1: In the approximation scenario all columns or rows belonging to a forbidden submatrix are deleted, whereas in the fixed-parameter setting a search tree algorithm branches recursively into several subcases, deleting in each case one of the columns or rows of the forbidden submatrix.

To show the performance guarantees of the thus derived algorithms, observe that a $(*, \Delta)$-matrix cannot contain submatrices of types M_{II_k} and M_{III_k} with arbitrarily large sizes. Therefore, the main difficulty is that every problem instance can contain submatrices of type M_{I_k} of unbounded size—the approximation factor or the number of cases to branch into would therefore not be bounded from above by Δ. To overcome this difficulty, we use the following two-phase approach:

1. Destroy only those forbidden submatrices that belong to a certain *finite* subset X of T (and whose sizes are upper-bounded, therefore).

Originally published in *Journal of Computer and System Sciences*. Elsevier B. V. To appear.

2. Solve Min-COS-C or Min-COS-R for each component of the resulting matrix. According to Corollary 2.1, these solutions can be combined into a solution for the whole input matrix.

The finite set $X \subseteq T$ is specified in the following theorem, the main structural contribution of this work. The technical proof is presented in Section 4.

Theorem 3.1. *Let* $X := \{M_{\mathrm{I}_k} \mid 1 \leq k \leq \Delta - 1\} \cup \{M_{\mathrm{II}_k} \mid 1 \leq k \leq \Delta - 2\} \cup \{M_{\mathrm{III}_k} \mid 1 \leq k \leq \Delta - 1\} \cup \{M_{\mathrm{IV}}, M_{\mathrm{V}}\}$*. If a* $(*, \Delta)$*-matrix* M *contains none of the matrices in* X *as a submatrix, then each component of* M *has the Circ1P.*

Now, to derive approximation and fixed-parameter algorithms, there remain two fundamental challenges:

1. Efficiently find a matrix from X, if existing.
2. Transform a matrix with Circ1P into a matrix with C1P.

After giving the proof for Theorem 3.1 in Section 4, we will address these two points separately in Section 5 and Section 6. A summary of our algorithmic results for Min-COS-C and Min-COS-R follows in Section 7.

4. Proof of the Main Structural Theorem

In this section, we prove Theorem 3.1.

Proof. We prove Theorem 3.1 by contraposition. More precisely, we show that if a component of a $(*, \Delta)$-matrix M does not have the Circ1P, then this component contains a submatrix in X. To this end, let A be a component of M not having the Circ1P. Then, by Corollary 2.2, there must be a column c of A such that the matrix A', resulting from A by complementing those rows that have a 1 in column c, does not have the C1P and, therefore, contains one of the submatrices in T (Theorem 2.1). In the following, we will make a case distinction based on which of the forbidden submatrices in T is contained in A' and which rows of A have been complemented, and show that in each case the matrix A contains a forbidden submatrix from X.

We denote the forbidden submatrix contained in A' with B' and the submatrix of A that corresponds to B' with B. Note that the matrix A' must contain a 0-column due to the fact that all 1s in column c have been complemented. Since no forbidden submatrix in T contains a 0-column, column c cannot belong to B' and, hence, not to B. We call c the *complementing column* of A.

When referencing to row or column indices of B', we will always assume that the rows and columns of B' are ordered as shown in Fig. 2.

Case 1: The submatrix B' is isomorphic to M_{IV}.

If no row of B has been complemented, then $B = B'$, and A also contains a submatrix M_{IV}, which belongs to X.

If exactly one of the first three rows of B has been complemented such that the resulting matrix is isomorphic to M_{IV}, then B contains one 0-column, and B

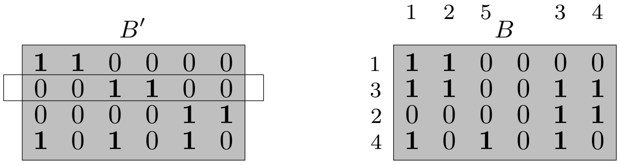

Figure 3: Illustration for Case 1 in the proof of Theorem 3.1. Complementing the second row of an M_{IV} (left side) generates an M_{V} (right side). (The rows and columns of the M_{V} are labeled with numbers according to the ordering of the rows and columns of the M_{V} in Fig. 2.) Note that complementing the fourth row of the matrix on the right side does not affect the existence of an M_{V}.

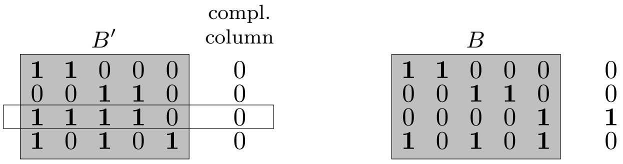

Figure 4: Illustration for Case 2 in the proof of Theorem 3.1. Suppose that only the third row of B is complemented. Then B together with the complementing column forms an M_{IV}.

without the 0-column forms an M_{V}, independent of whether the fourth row of B also has been complemented (see Fig. 3 for an example). Again, we have shown that A contains a submatrix from X.

If two or three of the first three rows of B have been complemented, then A contains an $M_{\mathrm{I}_1} \in X$ as a submatrix: Assume, for instance, that the first two rows have been complemented. If the fourth row has also been complemented, then there is an M_{I_1} consisting of the rows r_1, r_2, r_4 and the columns c_2, c_4, c_5 of B. Otherwise, there is an M_{I_1} consisting of the rows r_1, r_2, r_4 and the columns c_1, c_3, c_6 of B.

Case 2: The submatrix B' is isomorphic to M_{V}.

Analogously to Case 1 we can make a case distinction on which rows of A have been complemented, and in every subcase we can find a forbidden submatrix from X in A. In some of the subcases the forbidden submatrix can only be found in A if in addition to B also the complementing column of A is considered. We will present only one representative example for all subcases of Case 2: If only the third row of B has been complemented, then the complementing column of A contains a 0 in all rows that belong to B except for the third. Then B forms an M_{IV} together with the complementing column of A (see Fig. 4).

Case 3: The submatrix B' is isomorphic to M_{I_k} with $k \leq \Delta - 1$.

Subcase 3.1: No row of B has been complemented. Then $B = B'$, and A also contains a submatrix M_{I_k}.

Subcase 3.2: Exactly one row of B has been complemented. Then, together with the complementing column of A, the matrix B forms an M_{III_k}.

Subcase 3.3: At least two, but not all rows of B have been complemented.

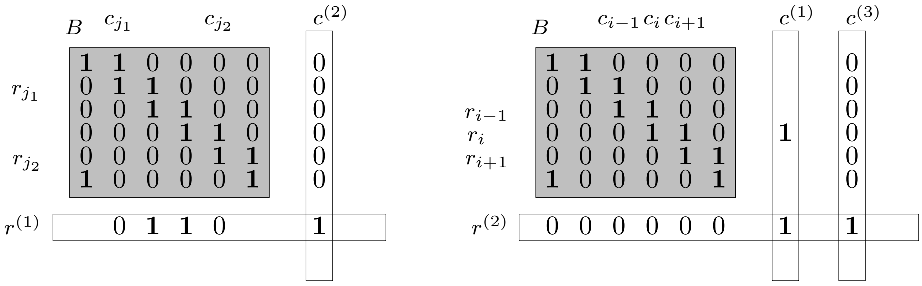

Figure 5: Illustration of Case 4 in the proof of Theorem 3.1.

If $k = 1$, then B contains a 0-column, and B with the 0-column deleted forms an M_{I_1} together with the complementing column of A. Otherwise, let $r_i, r_{i'}$ with $i' > i + 1$ be two complemented rows where no row $r_{i''}$ with $i < i'' < i'$ has been complemented. (We can assume that two such rows r_i and $r_{i'}$ exist because we can permute the rows and columns belonging to B' in an appropriate way due to the symmetry of B'.) If $i' = i + 2$, then the rows r_i, r_{i+1}, r_{i+2} and columns c_{i+1}, c_{i+2} of B form an M_{I_1} together with the complementing column of A. Otherwise, the rows $r_i, \ldots, r_{i'}$ and columns $c_{i+1}, \ldots, c_{i'}$ of B form an $M_{\mathrm{II}_{i'-i-2}}$ together with the complementing column of A. Note that $M_{\mathrm{II}_{i'-i-2}} \in X$ because $i' - i - 2 \leq k - 1 \leq \Delta - 2$.

Subcase 3.4: All rows of B have been complemented. If $k = 1$, then B forms an M_{III_1} together with the complementing column of A; if $k = 2$, then B forms an M_{I_2}; otherwise, there is an M_{I_1} consisting of the rows r_1, r_2, r_4 and the columns c_1, c_3, c_4 of B.

Case 4: The submatrix B' is isomorphic to M_{I_k} with $k \geq \Delta$.

Then no row of B has been complemented, because otherwise there would be a row in A that contains more than Δ 1s (note that the complementing column of A contains a 1 in every row that is complemented). Therefore, $B = B'$, and A also contains an M_{I_k}—but note that $k \geq \Delta$ and, therefore, $M_{\mathrm{I}_k} \notin X$.

Let c be the complementing column of A. Since no row of B has been complemented, the column c contains no 1 in a row that belongs to B—hence, the distance between c and B is greater than 1. However, column c must be connected to B due to the definition of a component, and, therefore, there must be a shortest path from c to B.

Now, make a case distinction on the parity of the distance between c and B.

Subcase 4.1: The distance between c and B is even. Then there is a shortest path $c^{(0)}, r^{(1)}, c^{(2)}, \ldots, c$ in A between B and c with $c^{(0)} \in B$. (If the distance between c and B is two, then $c = c^{(2)}$.) Note that, since the distance between c and B is even, the line $c^{(0)}$ must be a column. This means that the row $r^{(1)}$ does not belong to B, but has a 1 in a column that belongs to B and a 1 in column $c^{(2)}$. Column $c^{(2)}$ does neither belong to B nor does it have a 1 in a row that belongs to B. This constellation is displayed in the left part of Fig. 5. Because of $k \geq \Delta$, the matrix B has at least $\Delta + 2$ columns, and at least three columns of B must

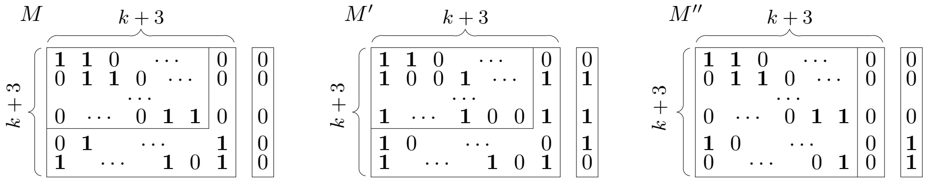

Figure 6: An illustration of the claim used in Case 5 of the proof of Theorem 3.1. Matrix M is composed of an M_{II_k} and a 0-column. Complementing rows r_2, r_{k+1}, and r_{k+2} of M leads to the matrix M'. Complementing the rows of M' that have a 1 in column c_{k+3}, namely, r_2, r_{k+1}, and r_{k+3}, transforms M' to matrix M'' which contains an $M_{\mathrm{I}_{k+1}}$ and a 0-column c_{k+3}.

contain a 0 in row $r^{(1)}$. Without loss of generality, let c_{j_1} and c_{j_2} with $j_2 > j_1+1$ be two columns containing a 0 in row $r^{(1)}$ such that all entries of row $r^{(1)}$ between c_{j_1} and c_{j_2} are 1s. (We can assume that such two columns c_{j_1} and c_{j_2} exist due to the symmetry of B'.) Then there is an $M_{\mathrm{III}_{j_2-j_1-1}}$ consisting of the rows $r_{j_1}, \ldots, r_{j_2-1}, r^{(1)}$ and columns $c_{j_1}, \ldots, c_{j_2}, c^{(2)}$. Since there can be at most Δ 1s in a row, we have $j_2-j_1-1 \leq \Delta-1$, and, therefore, $M_{\mathrm{III}_{j_2-j_1-1}} \in X$.

Subcase 4.2: The distance between c and B is odd. Then there is a shortest path $r^{(0)}, c^{(1)}, r^{(2)}, c^{(3)}, \ldots, c$ in A between B and c with $r^{(0)} \in B$. (If the distance between c and B is three, then $c = c^{(3)}$.) This means that the column $c^{(1)}$ does not belong to B, but it has a 1 in a row $r_i = r^{(0)}$ that belongs to B. (We can assume that $i > 1$ and $i < k+2$ due to the symmetry of B'.) Row $r^{(2)}$ does neither belong to B nor does it have a 1 in a column that belongs to B, but it has 1s in the columns $c^{(1)}$ and $c^{(3)}$. Column $c^{(3)}$ neither belongs to B nor does it have a 1 in a row that belongs to B. This constellation is depicted in the right part of Fig. 5. If column $c^{(1)}$ contains a 0 in row r_{i-1} as well as in row r_{i+1}, then there is an M_{IV} consisting of the rows $r_{i-1}, r_{i+1}, r^{(2)}, r_i$ and columns $c_{i-1}, \ldots, c_{i+2}, c^{(1)}$, $c^{(3)}$. If column $c^{(1)}$ contains a 1 in at least one of the rows r_{i-1} and r_{i+1}, say in r_{i-1}, then there is an M_{III_1} consisting of the rows $r_{i-1}, r_i, r^{(2)}$ and columns $c_{i-1}, c_{i+1}, c^{(1)}, c^{(3)}$.

Case 5: The submatrix B' is isomorphic to M_{II_k} with $k \geq 1$.

Here, we re-use the argumentation for A' containing an $M_{\mathrm{I}_{k+1}}$ (Case 3 and Case 4), since the matrix type M_{II_k} is closely related to M_{I_k}, as shown in the following claim.

Claim: For an integer $k \geq 1$, let M be a $(k+3) \times (k+4)$-matrix composed of an M_{II_k} and an additional 0-column, and let M' be any matrix resulting from M by complementing a subset of its rows. Then, complementing all rows of M' that have a 1 in column c_{k+3} results in a matrix containing $M_{\mathrm{I}_{k+1}}$ and an additional 0-column.

Proof of the claim: Let $R \subseteq \{1, 2, \ldots, k+3\}$ be the set of the indices of the rows that have been complemented in M in order to form M'. After complementing the rows r_i with $i \in R$ in M, the column c_{k+3} of M' contains 1s in all rows r_i with $i \in (\{1, \ldots, k+1\} \cap R) \cup (\{k+2, k+3\} \setminus R)$. It is easy to see that complementing these rows in M' results in the described matrix, proving the claim. See Fig. 6 for an illustration of the claim.

We return to the proof of Case 5. The matrix B' together with a 0-column has been created by complementing a subset of the rows belonging to B. Applying the above claim, regarding B' together with the 0-column as the matrix M mentioned in the claim, shows that there is a column c_j in A such that complementing all rows that contain a 1 in column c_j results in an $M_{\mathrm{I}_{k+1}}$ and a 0-column. Then A must contain a submatrix from X as we have shown in Case 3 and Case 4.

Case 6: The submatrix B' is isomorphic to M_{III_k} with $k \geq 1$.

Similarly to Case 5, this case can be reduced to Case 3 or Case 4 by applying the following claim, which reveals the relationship between matrix types M_{III_k} and M_{I_k}. This claim can be proven in analogy to the claim in Case 5.

Claim: For an integer $k \geq 1$, let $M = M_{\mathrm{III}_k}$, and let M' be any matrix resulting from M by complementing a subset of its rows. Then, complementing all rows of M' that have a 1 in column c_{k+3} results in a $(k+2) \times (k+3)$-matrix containing M_{I_k} and an additional 0-column. ☐

5. Fast Detection of Small Forbidden Submatrices

The algorithms based on the approach described in Section 3 search in every step of the first phase for a forbidden submatrix from the set X specified in Theorem 3.1. Hence, how to efficiently detect these forbidden submatrices is a crucial issue concerning the running times of these algorithms. Herein, note that the number of columns (in the case of Min-COS-C) or rows (in the case of Min-COS-R) of the submatrices from X is always bounded from above by a number depending on Δ, the maximum number of 1s per row. A straightforward exhaustive search would have to try $\Theta(m^{\Delta+1} \cdot n^{\Delta+2})$ possibilities. Here, we show how these small forbidden submatrices can be found in polynomial time with the degree of the polynomial *not* depending on Δ. For our search algorithms, we use a characterization of matrices having the C1P via asteroidal triples due to Tucker [44]. For a graph $G = (V, E)$, three vertices $u, v, w \in V$ form an *asteroidal triple* if between any two of them there exists a path in G that does not contain a vertex from the closed neighborhood of the third vertex.

Theorem 5.1 ([44, Theorem 6]). *A matrix M has the C1P iff its representing bipartite graph G_M does not contain an asteroidal triple whose three vertices correspond to columns of M.*

Using Theorem 5.1, a forbidden submatrix from T (see Theorem 2.1 and Fig. 2) in a given matrix M can be found as follows: For every vertex triple u, v, w in G_M corresponding to columns of M, determine the sum of the lengths of three shortest paths connecting u with v, u with w, and v with w, respectively, each time avoiding the closed neighborhood of the third vertex. If all three paths exist, then the vertices u, v, w form an asteroidal triple in G_M. Select a triple u, v, w where the sum is minimum compared to all other triples, and return the rows and columns of M that correspond to the vertices of the three

```
Input:  A binary matrix $M$.
Output: A submatrix $M'$ from $T$ occurring in $M$.
construct $G_M = (R \cup C, E)$;   // $R$ corresponds to rows and $C$ to columns
for every vertex $u \in C$: {
    $G^u := G[(R \cup C) \setminus N[u]]$;
    for every vertex $v \in C \setminus \{u\}$: {
        compute the lengths of all shortest paths in $G^u$ that start in $v$; }}
choose $u, v, w \in C$ such that $|P^u_G(v, w)| + |P^v_G(u, w)| + |P^w_G(u, v)|$ is minimum;
$V' := P^u_G(v, w) \cup P^v_G(u, w) \cup P^w_G(u, v)$;
$M'$ := the submatrix of $M$ whose rows and columns correspond to $V'$;
while $M'$ contains a row $r$ such that $M'$ without $r$ does not have the C1P
or a column $c$ such that $M'$ without $c$ does not have the C1P: {
    delete $r$ or $c$, respectively, from $M'$; }
return $M'$;
```

Figure 7: Algorithm for finding forbidden submatrices.

shortest paths computed for this triple. The returned submatrix must contain a submatrix from T; however, this procedure does not always return a submatrix of minimum size, because the sum of the lengths of the three paths computed for a triple u, v, w is not always the number of vertices in the union of the three paths—some vertices may be part of more than one path. In what follows, we start with analyzing the size of the returned matrix and show that it contains at most three more columns (five more rows) than a forbidden submatrix with minimum number of columns (rows). Later on, we will show how to find a submatrix with a minimum number of rows or a minimum number of columns or both (Theorem 5.2). Note that neither the known linear-time or polynomial-time algorithms (see Section 1) for deciding whether a given matrix has the C1P nor the known algorithms for finding an asteroidal triple in a graph (see [28]) output a minimum-size submatrix from T (and, thus, a forbidden submatrix from X) or a minimum-size induced subgraph containing an asteroidal triple.

Let $G = (V, E)$ be a graph and $u, v, w \in V$ be an asteroidal triple in G. With $P^u_G(v, w)$ we denote the vertex set of a shortest path in $G[V \setminus N[u]]$ between v and w (including v and w). Figure 7 contains the pseudocode of the algorithm behind the above approach. The following proposition gives an upper bound on the numbers of rows and columns of the submatrix returned by the algorithm.

Proposition 5.1. *Let M be a $(*, \Delta)$-matrix of size $m \times n$ that contains a forbidden $m' \times n'$-submatrix M' from T. Then the algorithm in Fig. 7 returns in $O(\Delta mn^2 + n^3)$ time a submatrix of M that belongs to T and has at most*

$$\begin{array}{ll}
m' \text{ rows and } n' \text{ columns} & \text{if } M' = M_{\mathrm{I}_k}, \\
m' \text{ rows and } n' \text{ columns} & \text{if } M' = M_{\mathrm{II}_k}, \\
m' + 3 \text{ rows and } n' + 2 \text{ columns} & \text{if } M' = M_{\mathrm{III}_k}, \\
m' + 5 \text{ rows and } n' + 3 \text{ columns} & \text{if } M' = M_{\mathrm{IV}}, \text{ and} \\
m' + 1 \text{ rows and } n' \text{ columns} & \text{if } M' = M_{\mathrm{V}}.
\end{array}$$

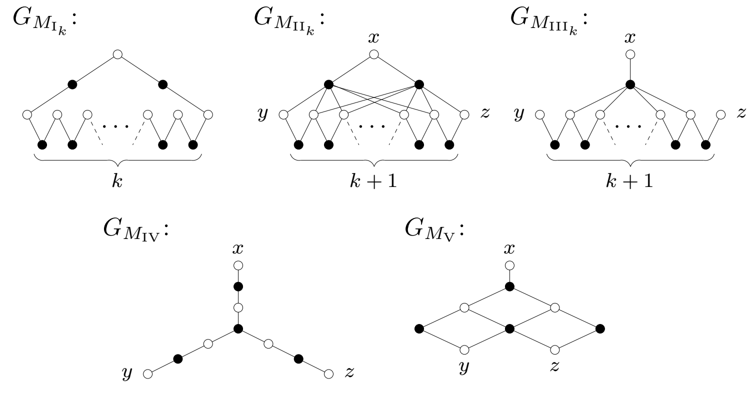

Figure 8: Representing graphs of the forbidden submatrices from T (Fig. 2) due to Tucker [44]. Black vertices correspond to rows, white vertices correspond to columns. The numbers k and $k+1$ refer to the number of black vertices in the lower parts of the first three graphs. In the case of matrix $M_{\mathrm{I}_k} \in T$, every triple of white vertices is an asteroidal triple. In all other cases, there is exactly one asteroidal triple consisting of white vertices; this triple is denoted by x, y, z.

Proof. By the above reasoning, the returned matrix M' clearly contains a submatrix from T. Furthermore, the lines 9 and 10 of the pseudocode in Fig. 7 ensure that M' is minimal; hence, the matrix M' must be one of the matrices from T.

Next, we prove the claimed row and column numbers of the returned matrix M'. Since M' does not have the C1P, the representing graph $G_{M'}$ of M' contains an asteroidal triple x, y, z corresponding to three columns of M' (Theorem 5.1). If $M' = M_{\mathrm{I}_k}$, then every triple of vertices corresponding to columns of M' is an asteroidal triple in $G_{M'}$. To see this, consider the first graph in Fig. 8, which shows the representing graphs of the forbidden submatrices from T: For every triple of white vertices, there is a path between any two of the vertices of the triple that avoids the closed neighborhood of the third. If $M' \neq M_{\mathrm{I}_k}$, then there is exactly one asteroidal triple in $G_{M'}$. This can be seen by considering the last four graphs in Fig. 8: The white vertices x, y, z form an asteroidal triple; any other triple of white vertices contains two vertices that are not connected by a path avoiding the closed neighborhood of the third. Let $p_{xyz} := |P^x_{G_{M'}}(y,z)| + |P^y_{G_{M'}}(x,z)| + |P^z_{G_{M'}}(x,y)|$. By considering the asteroidal triples in Fig. 8 one can verify that

$$\begin{aligned}
p_{xyz} &= 2k+7 && \text{if } M' = M_{\mathrm{I}_k},\\
p_{xyz} &= 2k+9 && \text{if } M' = M_{\mathrm{II}_k},\\
p_{xyz} &= 2k+13 && \text{if } M' = M_{\mathrm{III}_k},\\
p_{xyz} &= 21 && \text{if } M' = M_{\mathrm{IV}}, \text{ and}\\
p_{xyz} &= 13 && \text{if } M' = M_{\mathrm{V}}.
\end{aligned}$$

For example, if $M' = M_{\mathrm{III}_k}$, then $|P^x_{G_{M'}}(y,z)| = 2k+3$, $|P^y_{G_{M'}}(x,z)| = 5$, and $|P^z_{G_{M'}}(x,y)| = 5$, and, hence, $p_{xyz} = (2k+3)+5+5 = 2k+13$.

Let $u, v, w \in C$ be the vertices chosen in line 6 of the algorithm, and let $p_{uvw} := |P^u_G(v,w)| + |P^v_G(u,w)| + |P^w_G(u,v)|$. Clearly, $p_{uvw} \leq p_{xyz}$ because u, v, w are selected such that $|P^u_G(v,w)| + |P^v_G(u,w)| + |P^w_G(u,v)|$ is minimized. The returned submatrix consists of at most $(p_{uvw} - 3)/2 \leq (p_{xyz} - 3)/2$ rows because each of the vertices u, v, w is counted twice in p_{uvw} and because every second vertex in each of the vertex sets $P^u_G(v,w), P^v_G(u,w), P^w_G(u,v)$ corresponds to a column in M. It follows that the row number of the submatrix returned by the algorithm is upper-bounded by

$$\begin{array}{ll}
((2k+7)-3)/2 = k+2 = m' & \text{if } M' = M_{\mathrm{I}_k} \text{ (where } m' = k+2\text{)},\\
((2k+9)-3)/2 = k+3 = m' & \text{if } M' = M_{\mathrm{II}_k} \text{ (where } m' = k+3\text{)},\\
((2k+13)-3)/2 = k+5 = m'+3 & \text{if } M' = M_{\mathrm{III}_k} \text{ (where } m' = k+2\text{)},\\
(21-3)/2 = 9 = m'+5 & \text{if } M' = M_{\mathrm{IV}} \text{ (where } m' = 4\text{), and}\\
(13-3)/2 = 5 = m'+1 & \text{if } M' = M_{\mathrm{V}} \text{ (where } m' = 4\text{)}.
\end{array}$$

The number of columns in M' follows with a completely analogous argumentation.

To see the claimed running time, note that lines 2–5 can be executed in $O(n^2 \cdot (n + \Delta m))$ time by using breadth-first search in line 5: the number of vertices in C is n, and the input graph G^u for the breadth first search has $m+n$ vertices and at most Δm edges. For considering all triples u, v, w in line 6, the algorithm needs $O(n^3)$ time. The test in line 9 can be executed in linear time [6], that is, in $O(m' + n' + \Delta m')$ time, and, hence, the time needed for lines 9–10 is dominated by the time needed for lines 1–8. □

Next, we consider the consequences of Proposition 5.1 for the task of finding forbidden submatrices from X when solving Min-COS-C or Min-COS-R with the approach described in Section 3.

Corollary 5.1. *Let M be a $(*, \Delta)$-matrix of size $m \times n$.*

1. *If $\Delta = 3$ or $\Delta = 4$ and the algorithm in Fig. 7 does not find a forbidden submatrix from T consisting of at most 9 rows (columns), or*
2. *if $\Delta = 2$ or $\Delta \geq 5$ and the algorithm in Fig. 7 does not find a forbidden submatrix from T consisting of at most $\Delta + 4$ rows (columns),*

then M does not contain a forbidden submatrix from the set X specified in Theorem 3.1.

Proof. Assume that M contains a submatrix M' from X consisting of m' rows and n' columns. The number of rows and the number of columns of the matrix returned by the algorithm is upper-bounded by $k+2$ if $M' = M_{\mathrm{I}_k}$, by $k+3$ if $M' = M_{\mathrm{II}_k}$, by $k+5$ if $M' = M_{\mathrm{III}_k}$, by 9 if $M' = M_{\mathrm{IV}}$, and by 5 if $M' = M_{\mathrm{V}}$, as described in the proof of Proposition 5.1. Since, on the one hand, the matrices of the type M_{I_k} in X have $k \leq \Delta - 1$, the matrices of the type M_{II_k} in X have $k \leq \Delta - 2$, and the matrices of the type M_{III_k} in X have $k \leq \Delta - 1$, and, on the other hand, the matrix M can contain an M_{IV} as a submatrix only if $\Delta \geq 3$, it follows that the algorithm returns a matrix that has the claimed number of rows and columns. □

The number of columns and rows of the matrix returned by the algorithm in Fig. 7 is always close to the minimum number of columns or rows, respectively. However, if a submatrix from T shall be found that has exactly the minimum possible number of columns or rows, the algorithm in Fig. 7 is only useful in the case where this submatrix is of the type M_{I_k} or M_{II_k}. In the following, we present algorithms for finding a minimum-size submatrix of the type M_{III_k} and for finding submatrices of the types M_{IV} or M_{V}. The advantage of the algorithm in Fig. 7, however, is that of being faster.

In order to find a minimum-size submatrix of the type M_{III_k}, first observe that the representing graph of a matrix M_{I_k} is a hole. Hence, finding an induced M_{I_k} reduces to finding a minimum-size hole in a graph, a task which can be done in polynomial time (see below). For finding an induced M_{III_k}, we use the similarity between the matrix types M_{III_k} and M_{I_k}: the upper left part of an M_{III_k} is identical to the upper part of an M_{I_k}—the difference between an M_{III_k} and an M_{I_k} lies in the rightmost column and the bottommost row of the M_{III_k}. This similarity allows us to reduce the search for a minimum-size M_{III_k} to the search for a minimum-size hole. The connection between the matrix type M_{III_k} and holes in a graph can be formulated as follows.

Observation 5.1. *Let M be the $(k+2) \times (k+3)$-matrix that results from complementing the $(k+2)$-nd row of an M_{III_k}. Then the representing graph of M consists of an isolated vertex, corresponding to the $(k+3)$-rd column of M, and a chordless cycle.*

As a consequence of Observation 5.1, we get the following lemma.

Lemma 5.1. *Let M be a binary matrix and k be a positive integer. Then the following two statements are equivalent:*

1. *The matrix M contains an M_{III_k} as a submatrix.*
2. *There exist a column c_j and a row r_i in M with the following properties:*
    - *The row r_i has a 1 in column c_j.*
    - *If $\tilde{M}$ is the matrix consisting of*
        - *the row that results from complementing r_i and*
        - *all rows of M that have a 0 in column c_j,*

      then the representing graph of $\tilde{M}$ contains a chordless cycle H of length $2k+4$ that contains the vertex corresponding to the complemented row r_i.

Moreover, part (2) implies the following:

3. *The column c_j and the rows and columns corresponding to the vertices of the chordless cycle H together induce an M_{III_k} in M.*

```
Input:  A binary matrix $M$.
Output: A minimum-size induced submatrix of the type $M_{\mathrm{III}_k}$ occurring in $M$.
$M' := \emptyset$;
for every row $r_i$ of $M$: {
    for every column $c_j$ of $M$ having a 1 in row $r_i$: {
        $R_0 :=$ the set of rows having a 0 in column $c_j$;
        $\overline{r_i} := r_i$ complemented;
        $\tilde{M} :=$ the matrix consisting of $\overline{r_i}$ and all rows from $R_0$;
        $\tilde{G} :=$ the representing graph of $\tilde{M}$;
        search for a minimum-length hole in $\tilde{G}$ that contains the vertex
        corresponding to $\overline{r_i}$;
        if $H$ exists and $|V(H)| + 1 <$ number of rows and columns in $M'$: {
            $M' :=$ the submatrix of $M$ that is induced by the column $c_j$ and
                the rows and columns corresponding to the vertices of $H$; }}}
return $M'$;
```

Figure 9: Algorithm for finding a minimum-size submatrix of the type M_{III_k}.

Proof. **(1) ⇒ (2):** Let c_j be the column of M that contains the $(k+3)$-rd column of the M_{III_k} submatrix, and let r_i be the row of M that contains the $(k+2)$-nd row of the M_{III_k} submatrix. Then r_i is the only row of the M_{III_k} submatrix in M that has been complemented, and the claim follows from Observation 5.1.

(2) ⇒ (1) ∧ (3): This claim follows from the fact that the vertex corresponding to column c_j cannot be part of H, because in $\tilde{M}$ (after complementing row r_i) column c_j contains only 0s. Therefore, the submatrix of M that is induced by the column c_j and the rows and columns corresponding to the vertices of the chordless cycle H induce an M_{III_k}. □

Lemma 5.1 indicates how to find an induced M_{III_k} of minimum size: Try all combinations of one row r_i and one column c_j from M such that r_i contains a 1 in c_j. For each of these combinations, complement r_i, take all rows having a 0 in c_j, and search in the representing graph of the resulting matrix for the shortest hole having the properties mentioned in part (2) of Lemma 5.1—in particular, this hole must contain the vertex corresponding to r_i, and, since $k \geq 1$, it must have length $2k + 4 \geq 6$. Each representing graph to be considered has at most $m + n$ vertices and less than $\Delta m + n$ edges. Fig. 9 shows the pseudocode of this approach.

A shortest hole consisting of a given vertex r_i and at least five other vertices can be found as follows. Try all triples $(c_{j_1}, r_{i'}, c_{j_2})$ of vertices and search for the shortest hole on which the four vertices $r_i, c_{j_1}, r_{i'}, c_{j_2}$ appear consecutively. To find such a hole, c_{j_1} and $r_{i'}$ are deleted together with their neighbors except for r_i and c_{j_2}, and in the remaining graph a shortest path from c_{j_2} to r_i is sought. Since a shortest path in an unweighted graph can be found in linear time, we get the following result.

Proposition 5.2. *Let M be a $(*, \Delta)$-matrix of size $m \times n$. Then a minimum-size submatrix of the type M_{III_k} in M can be found in $O(\Delta^3 m^3 n + \Delta^2 m^2 n^2)$ time.*

To find a minimum-size submatrix of the type M_{IV} or M_{V}, we use an exhaustive search, which leads to the following result.

Proposition 5.3. *Let M be a $(*, \Delta)$-matrix of size $m \times n$. A submatrix of type M_{IV} can be found in $O(\Delta^3 m^2 n^3)$ time, and a submatrix of type M_{V} can be found in $O(\Delta^4 m^2 n)$ time.*

Proof. A matrix of type M_{IV} consists of six columns and four rows, and its fourth row contains three 1s. Since there are at most Δ 1s in every row of a $(*, \Delta)$-matrix M, the number of possibilities to select three columns from M that all contain a 1 in a specific row is bounded by $O(\Delta^3)$. Therefore, the idea for searching an M_{IV} in M is to iterate over all rows r_i of M and test whether M contains an M_{IV} in such a way that r_i forms the fourth row of the M_{IV}; this test can be performed by considering every triple of columns from M having a 1 in row r_i (there are $O(\Delta^3)$ such triples) in combination with every triple of columns from M having a 0 in row r_i (there are $O(n^3)$ such triples). For each of these combinations, check in $O(m)$ time whether every row of the matrix M_{IV} appears at least once in the submatrix induced by the selected columns. A submatrix of the type M_{V} can be found analogously. $\square$

The algorithm used in Proposition 5.3 leads to fast running times when searching a submatrix of one of the types M_{IV} and M_{V}, whereas the algorithm used in Proposition 5.2 efficiently finds a minimum-size submatrix of the type M_{III_k}. We combine the algorithms from Propositions 5.1, 5.3, and 5.2 to find a submatrix that is isomorphic to *any* of the submatrices from T and has a minimum number of rows, columns, rows and columns, or entries.

Theorem 5.2. *Let M be a $(*, \Delta)$-matrix of size $m \times n$. A forbidden submatrix from T in M that has a minimum number of rows can be found in $O(\Delta^3 m^2 n \cdot (m + n^2))$ time. Within the same time, one can also find a forbidden submatrix from T in M that has a minimum number of columns, a minimum number of rows and columns, or a minimum number of entries.*

Proof. The claimed running time can be obtained as follows: First, run the algorithm from Fig. 7 (Proposition 5.1), which finds a forbidden submatrix of "almost minimum" size, and let A be the returned submatrix. Second, run the algorithm from Fig. 9 (Proposition 5.2) to find a submatrix M_{III_k}, and let B be the submatrix found here. Third, run the algorithm of Proposition 5.3 two times, once for finding a submatrix M_{IV} and once for finding a submatrix M_{V}, and let C and D, respectively, be the found submatrices. Return the matrix with the minimum number of rows (columns, rows and columns, entries) out of A, B, C, and D.

The correctness of this approach is obvious: As shown in the proof of Proposition 5.1, if the forbidden submatrix from T in M with the minimum number of rows (columns, rows and columns, entries) is of the type M_{I_k} or M_{II_k}, then M does not contain a submatrix with less rows (columns, rows and columns, entries) than A. In all other cases, the forbidden submatrix from T in M with the minimum number of rows (columns, rows and columns, entries) must be one of B, C, and D. ☐

The only forbidden submatrices from T that can occur in a $(*, 2)$-matrix are the matrices M_{I_k}, $k \geq 1$, and M_{III_1}, which leads to the following corollary.

Corollary 5.2. *Let M be a $(*, 2)$-matrix of size $m \times n$. A forbidden submatrix from X in M that has a minimum number of columns (rows) can be found in $O(m^2n^2)$ time.*

6. From Circ1P to C1P

In this section, we consider the problems Min-COS-C and Min-COS-R restricted to input $(*, \Delta)$-matrices that have the Circ1P; these matrices arise in the second phase of the algorithmic skeleton described in Section 2.

To solve Min-COS-C (Min-COS-R) on a matrix M with the Circ1P, we first sort the columns of M in such a way that in every row the 1s appear consecutively in a circular sense (which, more precisely, means that in every row the 1s appear consecutively or the 0s appear consecutively or both). This can be done in linear time [26]. Min-COS-C (Min-COS-R) asks to delete a minimum-cardinality set of columns (rows) in such a way that in the resulting matrix the 1s can be placed consecutively in every row by permuting the columns. We will show that if the number n of columns is big enough compared to Δ, optimal solutions for Min-COS-C (Min-COS-R) have a special structure: It is always optimal to delete a set of columns (rows) in such a way that in the resulting matrix the 1s can be placed consecutively in every row *by a number of "cyclic shifts"*. In Section 6.1 we will prove this special structure of the optimal solutions, and in Sections 6.2 and 6.3 we show how to exploit it when solving Min-COS-C and Min-COS-R.

6.1. Circ1-Orderings and C1-Orderings

In what follows, it is helpful to imagine the matrices as wrapped around a vertical cylinder. Thus, a binary matrix M has the Circ1P if by permuting its columns a matrix M' can be obtained with the following property: If M' is wrapped around a vertical cylinder, then the 1s appear consecutively in every row. The matrix M' is said to have the *strong Circ1P*, and the corresponding column ordering is called a *Circ1-ordering* (see Fig. 10). If a binary matrix M has the strong Circ1P and, in addition, there is a column pair $(c_j, c_{\mathrm{succ}_n(j)})$ such that in every row r_i containing both 1s and 0s it holds that at most one of $m_{i,j}$ and $m_{i,\mathrm{succ}_n(j)}$ is 1, then we say that M has the *shifted strong C1P*, and its column ordering is called a *shifted C1-ordering* (see Fig. 10). The column pair $(c_j, c_{\mathrm{succ}_n(j)})$ is called a *C1-cut.*

It follows directly from these definitions that a binary matrix M has the Circ1P iff there is a Circ1-ordering for M's columns. Moreover, M has the C1P iff there is a shifted C1-ordering for M's columns, that is, iff there is a Circ1-ordering for M's columns that yields a C1-cut: If the column ordering $c_1, \ldots, c_n$

$$
\begin{array}{ccccc}
c_1 & c_2 & c_3 & c_4 & c_5 \\
\hline
0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\
\mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & \mathbf{1}
\end{array}
\qquad
\begin{array}{cc|ccc}
c_1 & c_2 & c_3 & c_4 & c_5 \\
\hline
0 & 0 & \mathbf{1} & \mathbf{1} & 0 \\
0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\
\mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\
\mathbf{1} & \mathbf{1} & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{ccccc}
c_3 & c_4 & c_5 & c_1 & c_2 \\
\hline
\mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\
0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\
0 & 0 & \mathbf{1} & \mathbf{1} & 0 \\
0 & 0 & 0 & \mathbf{1} & \mathbf{1}
\end{array}
$$

Figure 10: Left: A matrix with the strong Circ1P. Middle and right: Two matrices with the shifted strong C1P. The matrix on the right is obtained from the matrix in the middle by permuting the columns; for both matrices, the pair (c_2, c_3) is a C1-cut.

of a matrix is a Circ1-ordering and $(c_j, c_{\mathrm{succ}_n(j)})$ is a C1-cut, then the column ordering $c_{\mathrm{succ}_n(j)}, \ldots, c_n, c_1, \ldots, c_j$ places the 1s consecutively in every row of the resulting matrix (see Fig. 10). Intuitively speaking, wrapping M around a vertical cylinder, cutting the matrix on the cylinder vertically from top to bottom between c_j and $c_{\mathrm{succ}_n(j)}$, and unwrapping it from the cylinder places the 1s consecutively.

To prove the claimed structure of optimal solutions for MIN-COS-C and MIN-COS-R on matrices with the strong Circ1P, we show that if a matrix M has the C1P and the column number is big enough compared to Δ, then every Circ1-ordering for M's columns is a shifted C1-ordering; in other words, if the matrix has the strong Circ1P, then it also has the shifted strong C1P. To this end, we show that each Circ1-ordering for the columns of matrix can be obtained from a shifted C1-ordering by a series of column reversal operations, which do not destroy the shifted strong C1P.

Let $c_1, \ldots, c_n$ be the column ordering of a matrix. Given two column indices j_1, j_2, the operation $\mathsf{reverse}(c_{j_1}, c_{j_2})$ reverses the order of the columns between c_{j_1} and c_{j_2}: if $j_1 < j_2$, then $\mathsf{reverse}(c_{j_1}, c_{j_2})$ reverses the order of the columns $c_{j_1}, \ldots, c_{j_2}$, and if $j_1 > j_2$, then $\mathsf{reverse}(c_{j_1}, c_{j_2})$ reverses the order of the columns $c_{j_1}, \ldots, c_n, c_1, \ldots, c_{j_2}$. More intuitively, for reversing the columns from j_1 to j_2 in a matrix M, we first wrap M around a vertical cylinder, then apply the reverse operation as described, and finally cut the matrix on the cylinder vertically from top to bottom and unwrap it from the cylinder. If c_1 and c_n are still neighbors after reversing the columns, then this cut is made between c_1 and c_n; otherwise, there are two cases: if $j_2 = n$, then the cut is made to the left of c_1, and if $j_1 = 1$, then the cut is made to the right of c_n.

Definition 6.1 ([26])**.** *A subset C' of the columns of a matrix is called* uniform *in row r if all entries of row r in the columns of C' are the same. Let M be a matrix and let C be the set of its columns. A* circular module *of M is a subset $C' \subseteq C$ such that in every row r the subset C' is uniform in r or $C \setminus C'$ is uniform in r.*

Clearly, if a matrix M has the strong Circ1P, then applying the reverse operation to a set of columns that form a circular module does not destroy the strong Circ1P, that is, the operation transforms one Circ1-ordering into another one. However, there is an even stronger statement due to Hsu and

McConnell [26].

Theorem 6.1 ([26, Theorem 3.8])**.** *Let M be a matrix having the Circ1P. Then* every *Circ1-ordering for M's columns can be obtained by starting from an arbitrary Circ1-ordering and applying a sequence of* reverse *operations, each of them reversing a circular module.*

We can now state a useful relation between the Circ1-orderings and the shifted C1-orderings for the columns of matrices having the C1P. This observation is crucial for our algorithms solving Min-COS-R and Min-COS-C, since it implies that if n is big compared to Δ then it is optimal to delete a set of columns or rows, respectively, in such a way that the resulting matrix has the shifted strong C1P.

Lemma 6.1. *Let M be a $(*, \Delta)$-matrix of size $m \times n$, $n \geq 2\Delta - 1$, that has the C1P. Then every Circ1-ordering for M's columns is also a shifted C1-ordering.*

Proof. Since M has the C1P, its columns can be permuted such that the resulting matrix M' has the shifted strong C1P. By definition, then M' also has the strong Circ1P. We will prove the following claim:

Claim: Let M' be a matrix with the shifted strong C1P, and let M'' be a matrix obtained from M' by applying the reverse operation to an arbitrary circular module of M'. Then M'' has the shifted strong C1P.

Due to Theorem 6.1, this claim suffices to prove the lemma, because every Circ1-ordering for M's columns can be obtained from M' by a series of reverse operations, and by the claim, none of these operations destroys the shifted strong C1P.

Proof of the claim: Let C be the column set of M, and let $c_1, \ldots, c_n$ be the column ordering of M' (which is a shifted C1-ordering). Moreover, let $C' \subseteq C$ be the circular module of M' whose reversal leads to M''. Since M' has the shifted strong C1P, there is at least one C1-cut in M'. Without loss of generality, let (c_n, c_1) be this C1-cut, that is, there is no row r_i in M' with $m_{i,n} = 1$ and $m_{i,1} = 1$ and $m_{i,j} = 0$ for at least one $j \in \{2, \ldots, n-1\}$.

If C' does not contain c_1 and c_n, then (c_n, c_1) clearly is still a C1-cut after the reversal. Moreover, in this case M'' has also the strong Circ1P because, due to the definition of a circular module, the reversal of C' does not destroy this property. The shifted strong C1P and the existence of a C1-cut together imply the shifted strong C1P of M''. If C' contains both of c_1 and c_n, we can argue analogously because then (c_1, c_n) is a C1-cut in M''.

Now, assume that C' contains exactly one of c_1 and c_n, say c_1. Then $C' = \{c_1, \ldots, c_h\}$ with $h < n$, and M'' has the column ordering $c_h, \ldots, c_1, c_{h+1}, \ldots, c_n$. Assume for the sake of contradiction that none of (c_n, c_h) and (c_1, c_{h+1}) is a C1-cut in M''. Then there must be two rows r_{i_1} and r_{i_2} such that, on the one hand, $m_{i_1,n} = 1$ and $m_{i_1,h} = 1$, and, on the other hand, $m_{i_2,1} = 1$ and $m_{i_2,h+1} = 1$. Since (c_n, c_1) is a C1-cut in M', we have $m_{i_1,1} = 0$ and $m_{i_2,n} = 0$. Therefore, $m_{i_1,j} = 1$ for every $j \in \{h+1, \ldots, n-1\}$ and $m_{i_2,j} = 1$ for every $j \in \{2, \ldots, h\}$—otherwise, the set C' would not be a circular module.

Since there are at most Δ 1s in each row, $|\{c_h, \dots, c_n\}| \leq \Delta$ and, therefore, $h > n - \Delta \geq 2\Delta - 1 - \Delta = \Delta - 1$. For the same reason $|\{c_1, \dots, c_{h+1}\}| \leq \Delta$ and, therefore, $h \leq \Delta - 1$, contradicting $h > \Delta - 1$. Hence, at least one of (c_n, c_h) and (c_1, c_{h+1}) must be a C1-cut in M'', which implies the shifted strong C1P of M''. ☐

6.2. Solving MIN-COS-C *on Matrices with the Circ1P*

Here, we show how to use the results of Section 6.1 to solve MIN-COS-C on matrices with the Circ1P. We first give an upper bound on the solution size for MIN-COS-C on matrices having the Circ1P and then, exploiting Lemma 6.1, characterize the structure of optimal solutions for MIN-COS-C and show how to find them efficiently.

Lemma 6.2. *Let M be a $(*, \Delta)$-matrix that has the Circ1P. Then* MIN-COS-C *on input M can be solved by deleting at most Δ columns.*

Proof. Order the rows of M such that the resulting matrix M' has the strong Circ1P. Since each row of M' contains at most Δ 1s, the submatrix resulting from removing the leftmost Δ columns from M' has the (strong) C1P. ☐

Now, we show that there is always an optimal solution for MIN-COS-C with some nice structure, provided that the input matrix has the strong Circ1P and Δ is small enough compared to n.

Lemma 6.3. *Let M be a $(*, \Delta)$-matrix of size $m \times n$, $n \geq 3\Delta - 1$, that has the strong Circ1P, let $c_1, \dots, c_n$ be its column ordering, let the columns set C' be an optimal solution for* MIN-COS-C *on input M, and let M' be the matrix resulting from deleting C' from M.*[5] *Then,*

1. *M' has the shifted strong C1P and*
2. *in the matrix M, the columns from C' are consecutive in a circular way, and if c_α and c_β are the two columns to the left and to the right of C', (that is, $c_\alpha, c_\beta \notin C'$ and $c_{\mathrm{succ}_n(\alpha)}, c_{\mathrm{pred}_n(\beta)} \in C'$), then (c_α, c_β) is a C1-cut in M'.*

Proof. The idea behind the proof is as follows: First, show that the matrix M' fulfills the conditions of Lemma 6.1. Hence, by deleting C' from M, one obtains a matrix M' that does not only have the C1P, but also has the shifted strong C1P, which proves (1). Then, show that this fact implies (2).

The details are as follows. Let $c_{j_1}, \dots, c_{j_{n'}}$ be the columns of M', that is, the columns of M that do not belong to C'. Due to Lemma 6.2, $|C'| \leq \Delta$, and, therefore, M' has at least $2\Delta - 1$ columns. By Lemma 6.1, this implies that M' has the shifted strong C1P, because the strong Circ1P is preserved when deleting columns. This proves statement (1).

---

[5] When columns are deleted, the remaining columns retain the numbering scheme of the original matrix.

To prove statement (2), assume without loss of generality that $(c_{j_{n'}}, c_{j_1})$ is a C1-cut of M', that is, the column ordering $c_{j_1}, \ldots, c_{j_{n'}}$ places the 1s consecutively in every row. Suppose, for the sake of contradiction, that (2) does not hold, that is, there exists a column $c_x \in C'$ such that when M is wrapped around a vertical cylinder, the column c_x appears to the right of c_{j_1} and to the left of $c_{j_{n'}}$.

Let M'' be the matrix that results from M' by inserting the column c_x at its "old position", that is, M'' results from M by deleting the columns $C' \setminus \{c_x\}$. Clearly, M'' has the strong Circ1P because M has the strong Circ1P. Moreover, the insertion of c_x into M' does not affect the fact that $(c_{j_{n'}}, c_{j_1})$ is a C1-cut. Hence, the matrix M'' also has the shifted strong C1P. This means that $C' \setminus \{c_x\}$ is also a solution of Min-COS-C, contradicting the optimality of C' as a solution. □

By Lemma 6.3, the columns of an optimal solution C' are consecutive in *every* Circ1-ordering for M's columns. Hence, an optimal solution can easily be found.

Theorem 6.2. Min-COS-C*, restricted to* $(*, \Delta)$*-matrices of size* $m \times n$ *that have the Circ1P, can be solved in* $O((3\Delta)^{\min\{d,\Delta\}} \cdot \Delta m)$ *time if* $n < 3\Delta - 1$*, and in* $O(\Delta mn)$ *time otherwise, where* d *is the number of allowed column deletions.*

Proof. Let M be a $(*, \Delta)$-matrix of size $m \times n$ that has the Circ1P. Due to Lemma 6.2, an optimal solution for Min-COS-C on M has size at most Δ. If $n < 3\Delta - 1$, then an optimal solution can be found by trying all possibilities to delete at most $\min\{d, \Delta\}$ columns (there are $\binom{n}{\min\{d,\Delta\}} = O((3\Delta)^{\min\{d,\Delta\}})$ possibilities) and checking in $O(\Delta m + n)$ time [6] whether the resulting matrix has the C1P. If $n \geq 3\Delta - 1$, then assume that M has the strong Circ1P (a Circ1-ordering for M's columns can be found in $O(\Delta m + n)$ time [6]). Due to Lemma 6.3, there exists an optimal solution C' that is consecutive in the circular ordering of M and that is enclosed by the columns of a C1-cut in the matrix resulting from the deletion of C'. This solution can be found by checking, for every column pair $(c_j, c_{j'})$ with at most Δ columns lying between c_j and $c_{j'}$ in the circular ordering of M, whether the submatrix of M that consists of the columns $c_1, \ldots, c_j, c_{j'}, \ldots, c_n$ has the strong C1P with $(c_j, c_{j'})$ being a C1-cut. For such a check, simply test in $O(m)$ time whether for every row r_i at least one of $m_{i,j}$ and $m_{i,j'}$ is 0. □

6.3. *Solving* Min-COS-R *on Matrices with the Circ1P*

In the case of Min-COS-R, we cannot upper-bound the size of an optimal solution as we did in Lemma 6.2. However, Lemma 6.1 yields a characterization of optimal solutions for Min-COS-R that is very similar to the one given in Lemma 6.3 for Min-COS-C.

Lemma 6.4. *Let* M *be a* $(*, \Delta)$*-matrix of size* $m \times n$*,* $n \geq 2\Delta - 1$*, that has the strong Circ1P, let the set* R' *of rows be an optimal solution for* Min-COS-R *on input* M*, let* M' *be the matrix that results from deleting* R' *from* M*, and let* $c_1, \ldots, c_n$ *be the column ordering of* M *and* M'*. Then,*

1. *M' has the shifted strong C1P and*
2. *there is a C1-cut $(c_j, c_{\mathrm{succ}_n(j)})$ in M' such that*

$$R' = \{r_i \mid (1 \leq i \leq m) \wedge (r_i \text{ contains 0s and 1s}) \wedge (m_{i,j} = m_{i,\mathrm{succ}_n(j)} = 1)\}.$$

Proof. Lemma 6.1 implies that M' has the shifted strong C1P because M' obviously has the strong Circ1P. This proves (1). To prove (2), let $(c_j, c_{\mathrm{succ}_n(j)})$ be an arbitrary C1-cut in M'. On the one hand, due to the definition of a C1-cut, there can be no row in M' that contains 0s and 1s and that contains a 1 in both c_j and $c_{\mathrm{succ}_n(j)}$. Hence, all rows r_i with $m_{i,j} = 1$ and $m_{i,\mathrm{succ}_n(j)} = 1$ and $m_{i,j'} = 0$ for at least one j' must be part of R'. On the other hand, suppose, for the sake of contradiction, that there exists a row in R' that contains only 1s or only 0s or that does not contain a 1 in both c_j and $c_{\mathrm{succ}_n(j)}$. Then, reinserting this row into M' results in a matrix that still has the strong C1P—a contradiction to the optimality of R'. □

In analogy to Min-COS-C (Theorem 6.2), an optimal solution can now easily be found by exploiting Lemma 6.4.

Theorem 6.3. Min-COS-R, *restricted to $(*, \Delta)$-matrices of size $m \times n$ that have the Circ1P, can be solved in $O((2\Delta)^{2\min\{d,4\Delta^2\}} \cdot \Delta m)$ time if $n < 2\Delta - 1$, and in $O(mn)$ time otherwise, where d is the number of allowed row deletions.*

Proof. Let M be a $(*, \Delta)$-matrix of size $m \times n$ that has the Circ1P. If $n < 2\Delta - 1$, then first eliminate duplicate rows by assigning weights to the rows such that every row gets as weight the number of its occurrences, and by deleting all occurrences except for one of every row. The number of rows of the resulting matrix is bounded from above by $(2\Delta)^2$ because if M has the strong Circ1P, then every row can be described uniquely by the index of the first and last column containing a 1 in this row, which yields $(2\Delta)^2$ possibilities. The task is now to find a row set of minimum weight whose deletion yields the C1P. An optimal solution can be found by trying all possibilities to delete at most $\min\{d, (2\Delta)^2\}$ rows and checking in $O(\Delta m + n)$ time [6] whether the resulting matrix has the C1P; the number of possibilities to try is $\binom{(2\Delta)^2}{\min\{d,(2\Delta)^2\}}$.

If $n \geq 2\Delta - 1$, then assume that M has the strong Circ1P (a Circ1-ordering for M's columns can be found in $O(\Delta m + n)$ time [6]). Due to Lemma 6.4, an optimal solution can be found by counting for every column pair $(c_j, c_{\mathrm{succ}_n(j)})$ in $O(m)$ time the number of rows r_i with $m_{i,j} = 1$ and $m_{i,\mathrm{succ}_n(j)} = 1$ and $m_{i,j'} = 0$ for at least one j'; deleting these rows results in a matrix with C1-cut $(c_j, c_{\mathrm{succ}_n(j)})$. □

7. Algorithms for $(*, \Delta)$-Matrices

As sketched in the algorithmic skeleton of Section 3, our approximation algorithms for Min-COS-C and Min-COS-R consist of two phases: First, they search in every step for a matrix of the set X of forbidden submatrices given by

Table 2: Summary of results for MIN-COS-C and MIN-COS-R on $(*, \Delta)$-matrices.

Approximation algorithms

MIN-COS-C	Factor	Running time	based on
$\Delta = 2$	4	$O(m^2 n^3)$	Cor. 5.2, Lem. 6.2
$\Delta = 3$	6	$O(m^3 n^2 + m^2 n^4)$	Thm. 5.2, Lem. 6.2
$\Delta \geq 4$	Δ+2	$O(\Delta^3 m^3 n^2 + \Delta^3 m^2 n^4)$	Thm. 5.2, Lem. 6.2
$\Delta = 2, 5, 6, \ldots$	Δ+4	$O(\Delta mn^3 + n^4)$	Cor. 5.1, Lem. 6.2
$\Delta = 3, 4$	9	$O(mn^3 + n^4)$	Cor. 5.1, Lem. 6.2

MIN-COS-R	Factor	Running time	
$\Delta = 2$	3	$O(m^3 n^2)$	Cor. 5.2, Thm. 6.3
$\Delta \geq 3$	Δ+1	$O((2\Delta)^{8\Delta^2} \cdot \Delta m^2 + \Delta^3 m^4 n + \Delta^3 m^3 n^3)$	Thms. 5.2, 6.3
$\Delta = 2, 5, 6, \ldots$	Δ+4	$O((2\Delta)^{8\Delta^2} \cdot \Delta m^2 + \Delta m^2 n^2 + mn^3)$	Cor. 5.1, Thm. 6.3
$\Delta = 3, 4$	9	$O(m^2 n^2 + mn^3)$	Cor. 5.1, Thm. 6.3

Fixed-parameter algorithms

MIN-COS-C	Running time	based on
$\Delta = 2$	$O(4^d \cdot m^2 n^2)$	Cor. 5.2, Thm. 6.2
$\Delta = 3$	$O(6^d \cdot (m^2 n \cdot (m + n^2)))$	Thms. 5.2, 6.2
$\Delta \geq 4$	$O((\Delta{+}2)^d \cdot ((3\Delta)^{\min\{d,\Delta\}} \cdot \Delta dm + \Delta^3 m^3 n + \Delta^3 m^2 n^3))$	Thms. 5.2, 6.2
$\Delta = 2, 5, 6, \ldots$	$O((\Delta{+}4)^d \cdot ((3\Delta)^{\min\{d,\Delta\}} \cdot \Delta dm + \Delta mn^2 + n^3))$	Cor. 5.1, Thm. 6.2
$\Delta = 3, 4$	$O(9^d \cdot (mn^2 + n^3))$	Cor. 5.1, Thm. 6.2

MIN-COS-R	Running time	
$\Delta = 2$	$O(3^d \cdot m^2 n^2)$	Cor. 5.2, Thm. 6.3
$\Delta \geq 3$	$O((\Delta{+}1)^d \cdot ((2\Delta)^{2 \cdot \min\{d, 4\Delta^2\}} \cdot \Delta dm + \Delta^3 m^3 n + \Delta^3 m^2 n^3))$	Thms. 5.2, 6.3
$\Delta = 2, 5, 6, \ldots$	$O((\Delta{+}4)^d \cdot ((2\Delta)^{2 \cdot \min\{d, 4\Delta^2\}} \cdot \Delta dm + \Delta mn^2 + n^3))$	Cor. 5.1, Thm. 6.3
$\Delta = 3, 4$	$O(9^d \cdot (mn^2 + n^3))$	Cor. 5.1, Thm. 6.3

Theorem 3.1 and then delete all columns (rows) of the found submatrix. Since an optimal solution has to delete at least one column (row) of every forbidden submatrix from X, the approximation factor is bounded from above by the maximum number of columns (rows) of a submatrix found during this phase. Thereafter, due to Theorem 3.1, all components of the remaining matrix have the Circ1P. In case of MIN-COS-C, a solution of size at most Δ can be found for every component by permuting its columns such that the strong Circ1P is obtained and then deleting the first Δ columns (as shown in the proof of Lemma 6.2)—clearly, this yields a factor-Δ approximation for every component. The overall approximation factor is determined by the one achieved in the first phase of the algorithm. In case of MIN-COS-R, we do not have such a simple factor-Δ approximation for solving the problem on the components of the matrix resulting from the first phase. Hence, we use the approach of Theorem 6.3 for exactly solving MIN-COS-R on every component resulting from the first phase. Note that to derive polynomial running times for fixed Δ, we can ignore the term d in the running time of Theorem 6.3.

The fixed-parameter search tree algorithms look in every step for a forbidden submatrix of X and then branch on which column (row) belonging to the found submatrix shall be deleted. The solution for the resulting matrices without submatrices from X can be found without branching, see Theorem 6.2 (Theorem 6.3).

Theorem 7.1. Min-COS-C *and* Min-COS-R*, restricted to* $(*, \Delta)$*-matrices, have constant-factor approximation algorithms as shown in Table 2. Moreover,* Min-COS-C *and* Min-COS-R *are fixed-parameter tractable with respect to the parameter* d *denoting the number of allowed column deletions and row deletions, respectively. The running times are given in Table 2.*

Proof. In the case of the approximation algorithms, the approximation factor is determined by the number of columns (rows) of the submatrices found in the first phase of the algorithm. If the algorithm in Fig. 7 is used for searching forbidden submatrices from X, then the column number (row number) is determined by Corollary 5.1. If, otherwise, the algorithm behind Theorem 5.2 (or Corollary 5.2 in the case of $\Delta = 2$) is used, then the column number (row number) is equal to the maximum taken over the column numbers (row numbers) of the matrices in X. Since at most n columns (m rows) can be deleted, the running time for every algorithm is n times (m times) the time needed for searching a forbidden submatrix (see Proposition 5.1, Theorem 5.2, and Corollary 5.2) plus n times (m times) the time needed for approximating Min-COS-C (solving Min-COS-R) on a component that has the Circ1P.

In case of the search-tree algorithms, the number of branches depends on the maximum number of columns (rows) of a forbidden submatrix found during the first phase of the algorithm, which destroys all submatrices from X, and is either determined by Corollary 5.1 or by the maximum taken over the column numbers (row numbers) of the matrices in X—depending on which algorithm is used for searching the forbidden submatrices. The time needed in each node of the search tree is given by the time needed to search for a submatrix from X (see Proposition 5.1, Theorem 5.2, and Corollary 5.2) plus, in the case that no submatrix from X was found, the time needed for solving Min-COS-C (Min-COS-R) on at most d components that have the Circ1P. ☐

8. $(*, 2)$- and $(2, *)$-Matrices

Min-COS-R and Min-COS-C remain NP-hard on $(*, 2)$-matrices and $(2, *)$-matrices [8, 42]. However, for $(*, 2)$-matrices a fruitful interaction with natural graph problems can be exploited because then the 0/1-matrices have an interpretation as graphs. This is the central observation used for our algorithmic results concerning $(*, 2)$-matrices. Since most of the observations used in this section are fairly canonical, we only summarize our findings in an informal way and refer to the first author's PhD thesis [8] for any technical details.

We start with enumerating the $(*, 2)$-matrix problems together with their corresponding graph problems:

- Min-COS-C on $(*, 2)$-matrices is equivalent to the problem of finding in an undirected graph a minimum-cardinality set of vertices whose removal leaves a union of vertex disjoint paths [42]. (Note that the removal of vertices corresponds to the deletion of columns.)

- MIN-COS-R on $(*, 2)$-matrices is equivalent to the problem of finding in an edge-weighted undirected graph a minimum-weight set of edges whose removal leaves a union of vertex disjoint paths [8]. (Note that the removal of edges corresponds to the deletion of rows.)

In the case of $(2, *)$-matrices, only MIN-COS-C has a direct characterization as a graph problem: Given an edge-weighted undirected graph, find a minimum-weight set of edges whose removal leaves a union of vertex disjoint caterpillars [8, 42]. (A caterpillar is a tree in which every non-leaf vertex has at most two non-leaf neighbors.) For obtaining algorithms for *both* MIN-COS-C and MIN-COS-R on $(2, *)$-matrices, we do not use the graph interpretation of these matrices, but the following approach: In a way very much analogous to Theorem 3.1 one can show that if a $(2, *)$-matrix M without identical columns does not contain an M_{IV} or an M_{I_1} and does not have the C1P, then M contains pairwise disjoint M_{I_k} matrices (and no other forbidden submatrices from T). This characterization leads to almost straightforward search tree algorithms and approximation algorithms (see [8] for the details).

Altogether, we state the following results for $(*, 2)$- and $(2, *)$-matrices.

Theorem 8.1. *1. On $(*, 2)$-matrices,* MIN-COS-C *has a problem kernel consisting of a matrix with $O(d^2)$ rows and columns, and* MIN-COS-R *has a problem kernel consisting of a matrix with less than $9d$ different rows, less than $8d$ columns, and an overall number of at most $9d^2 + 9d$ rows.*

2. On $(2, *)$-matrices, MIN-COS-C *can be solved with a search tree algorithm running in $O(6^d \cdot \min\{m^4 n, m^2 n^3\})$ time, and* MIN-COS-R *can be solved with a search tree algorithm running in $O(4^d \cdot \min\{m^4 n, m^2 n^3\})$ time. Correspondingly, there is a factor-6 polynomial-time approximation algorithm for* MIN-COS-C *and a factor-4 polynomial-time approximation algorithm for* MIN-COS-R*.*

We close this section with a few comments on the proof of Theorem 8.1; the details can be found in [8]. As to (1) in Theorem 8.1, note that simple polynomial-time executable data reduction rules suffice to show the kernel [8] (see also [15] for related graph problems). As to (2) in Theorem 8.1, note that the exponential base 6 relates to the at most six columns to be deleted from forbidden M_{IV}- and M_{I_1}-submatrices; similarly, the exponential base 4 relates to the at most four rows in these submatrices. Finally, we remark that MIN-COS-C on $(*, 2)$-matrices without identical columns is equivalent to the graph problem 2-LAYER PLANARIZATION; see [13, 14, 40, 41] for results on this problem.

9. Outlook

Our results mainly focus on MIN-COS-C and MIN-COS-R with no restriction on the number of 1s in the columns; similar studies would be desirable for the case that we have no restriction for the rows. Moreover, it should be investigated whether the running times for Min-COS-R and Min-COS-C (see Table 2) can be improved. In particular, we think that approximating Min-COS-R with a factor of $\Delta + 1$ should be possible within a running time that is polynomial in the input size and has no exponential factor depending on Δ. An important research direction is to consider the problem Min-CO-1E (flipping of 1-entries). We conjecture that for $(*, \Delta)$-matrices the presented approximation and fixed-parameter tractability results should extend to Min-CO-1E—however, we could not prove that. Only for $\Delta = 2$ we have algorithmic results simply based on the equivalence to Min-COS-R in this case.

References

[1] E. Althaus, S. Canzar, M. R. Emmett, A. Karrenbauer, A. G. Marshall, A. Meyer-Baese, and H. Zhang. Computing H/D-exchange speeds of single residues from data of peptic fragments. In *Proceedings of the 23rd ACM Symposium on Applied Computing (SAC '08)*, pages 1273–1277. ACM Press, 2008.

[2] J. E. Atkins, E. G. Boman, and B. Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM Journal on Computing*, 28(1):297–310, 1998.

[3] J. E. Atkins and M. Middendorf. On physical mapping and the consecutive ones property for sparse matrices. *Discrete Applied Mathematics*, 71(1–3):23–40, 1996.

[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer, 1999.

[5] J. J. Bartholdi, III, J. B. Orlin, and H. D. Ratliff. Cyclic scheduling via integer programs with circular ones. *Operations Research*, 28(5):1074–1085, 1980.

[6] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

[7] T. Christof, M. Oswald, and G. Reinelt. Consecutive ones and a betweenness problem in computational biology. In *Proceedings of the 6th International Conference on Integer Programming and Combinatorial Optimization (IPCO '98)*, volume 1412 of *LNCS*, pages 213–228. Springer, 1998.

[8] M. 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.

[9] M. Dom. Algorithmic aspects of the consecutive-ones property. *Bulletin of the European Association for Theoretical Computer Science*, 2009. To appear.

[10] M. Dom, J. Guo, and R. Niedermeier. Approximability and parameterized complexity of consecutive ones submatrix problems. In *Proceedings of the 4th Annual Conference on Theory and Applications of Models of Computation (TAMC '07)*, volume 4484 of *LNCS*, pages 680–691. Springer, 2007.

[11] M. Dom and R. Niedermeier. The search for consecutive ones submatrices: Faster and more general. In *Proceedings of the 3rd Algorithms and Complexity in Durham (ACiD '07) Workshop*, volume 9 of *Texts in Algorithmics*, pages 43–54. College Publications, 2007.

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[13] V. Dujmovic, M. R. Fellows, M. T. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. A. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood. A fixed-parameter approach to 2-Layer Planarization. *Algorithmica*, 45(2):159–182, 2006.

[14] H. Fernau. Two-layer planarization: Improving on parameterized algorithmics. In *Proceedings of the 31st Conference on Current Trends in Theory and Practice of Informatics (SOFSEM '05)*, volume 3381 of *LNCS*, pages 137–146. Springer, 2005.

[15] H. Fernau. Parameterized algorithmics for linear arrangement problems. *Discrete Applied Mathematics*, 156(17):3166–3177, 2008.

[16] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.

[17] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–855, 1965.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[19] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.

[20] M. Habib, R. M. McConnell, C. Paul, and L. Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234(1–2):59–84, 2000.

[21] M. Hajiaghayi. Consecutive ones property. Manuscript, School of Computer Science, University of Waterloo, Canada, 2000.

[22] M. Hajiaghayi and Y. Ganjali. A note on the consecutive ones submatrix problem. *Information Processing Letters*, 83(3):163–166, 2002.

[23] R. Hassin and N. Megiddo. Approximation algorithms for hitting objects with straight lines. *Discrete Applied Mathematics*, 30:29–42, 1991.

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

[25] W.-L. Hsu. A simple test for the consecutive ones property. *Journal of Algorithms*, 43(1):1–16, 2002.

[26] W.-L. Hsu and R. M. McConnell. PC trees and circular-ones arrangements. *Theoretical Computer Science*, 296(1):99–116, 2003.

[27] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2 - \epsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[28] E. Köhler. Recognizing graphs without asteroidal triples. *Journal of Discrete Algorithms*, 2(4):439–452, 2004.

[29] N. Korte and R. H. Möhring. An incremental linear-time algorithm for recognizing interval graphs. *SIAM Journal on Computing*, 18(1):68–81, 1989.

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

[31] D. Kratsch, R. M. McConnell, K. Mehlhorn, and J. Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. *SIAM Journal on Computing*, 36(2):326–353, 2006.

[32] R. M. McConnell. A certifying algorithm for the consecutive-ones property. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, pages 768–777. ACM Press, 2004.

[33] S. Mecke, A. Schöbel, and D. Wagner. Station location – complexity and approximation. In *Proceedings of the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS '05)*. IBFI Dagstuhl, Germany, 2005.

[34] S. Mecke and D. Wagner. Solving geometric covering problems by data reduction. In *Proceedings of the 12th Annual European Symposium on Algorithms (ESA '04)*, volume 3221 of *LNCS*, pages 760–771. Springer, 2004.

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

[36] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[37] M. Oswald and G. Reinelt. The weighted consecutive ones problem for a fixed number of rows or columns. *Operations Research Letters*, 31(3):350–356, 2003.

[38] M. Oswald and G. Reinelt. The simultaneous consecutive ones problem. *Theoretical Computer Science*, 410(21–23):1986–1992, 2009.

[39] N. Ruf and A. Schöbel. Set covering with almost consecutive ones property. *Discrete Optimization*, 1(2):215–228, 2004.

[40] M. Suderman. *Layered Graph Drawing.* PhD thesis, School of Computer Science, McGill University Montréal, Canada, 2005.

[41] M. Suderman and S. Whitesides. Experiments with the fixed-parameter approach for two-layer planarization. In *Proceedings of the 11th International Symposium on Graph Drawing (GD '03)*, volume 2912 of *LNCS*, pages 345–356. Springer, 2003.

[42] J. Tan and L. Zhang. The consecutive ones submatrix problem for sparse matrices. *Algorithmica*, 48(3):287–299, 2007.

[43] A. C. Tucker. Matrix characterizations of circular-arc graphs. *Pacific Journal of Mathematics*, 2(39):535–545, 1971.

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

[45] V. V. Vazirani. *Approximation Algorithms.* Springer, 2001.

[46] A. F. Veinott and H. M. Wagner. Optimal capacity scheduling. *Operations Research*, 10:518–547, 1962.

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