Fixed-Parameter Tractability of the Simultaneous Consecutive Ones Submatrix & Editing Problems.

Rani M. R.¹ and R. Subashini²

1 Department of Computer Science & Engineering.
National Institute of Technology, Calicut
 rani_p150067cs@nitc.ac.in
2 Department of Computer Science & Engineering.
National Institute of Technology, Calicut
suba@nitc.ac.in

Abstract

A binary matrix has the consecutive ones property (C1P) for rows (columns) if there is a permutation of its columns (rows) that arranges the ones consecutively in all the rows (columns). A binary matrix has the simultaneous consecutive ones property (SC1P) if it has both the C1P for rows and columns. We consider the following two categories of problems: Simultaneous Consecutive Ones Submatrix (SC1S) and Simultaneous Consecutive Ones Editing (SC1E) [17] in a similar spirit as that of Consecutive Ones Submatrix [8] and Consecutive Ones Editing [5] problems. For a given binary matrix M, Consecutive Ones Submatrix problem decides if there exists a submatrix of M that satisfies the C1P for columns whereas SC1E problem deals with flipping a minimum number of 1s to obtain the SC1P for rows. We consider the parameterized versions of SC1S and SC1E problems with k as the parameter and are defined as follows. Given a binary matrix M, Consecutive Ones Submatrix problem decides if there is a submatrix of M that satisfies the C1P for rows and Consecutive Ones Editing problem decides if there is a set of 1-entries of size at most k in M whose flipping results in a matrix with the C1P for rows. SC1S problems focus on deleting a minimum number of rows (columns) to establish the SC1P whereas SC1E problem deals with flipping a minimum number of 1s to obtain the SC1P. We consider the parameterized versions of SC1S and SC1E problems with k as the parameter and are defined as follows. Given a binary matrix M and a positive integer k, k-SC1S-R (k-SC1S-C) problem decides whether there exists a set of rows (columns) of size at most k whose deletion results in a matrix with the SC1P. The k-SC1P-1E problem decides whether there exists a set of 1-entries of size at most k whose flipping results in a matrix with the SC1P. We show that k-SC1S-R, k-SC1S-C and k-SC1P-1E are fixed-parameter tractable on (2, *)-matrices and (*, 2)-matrices. We observe that using our algorithm, k-SC1S-R, k-SC1S-C and k-SC1P-1E can be approximated in polynomial-time on (2, *)-matrices and (*, 2)-matrices.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems, G.2.2 Graph Theory, I.1.2 Algorithms

Keywords and phrases Consecutive Ones Property, Simultaneous Consecutive Ones Property, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Binary matrices having simultaneous consecutive ones property are fundamental in recognizing biconvex graphs [19], identifying blocks of matrices (in applications arising from integer linear programming) [10] and finding clusters from metabolic networks [11]. A binary matrix has the consecutive ones property (C1P) for rows (columns) [7], if there is a permutation of its columns (rows) that arranges the ones consecutively in all the rows (columns). A binary matrix has the simultaneous consecutive ones property (SC1P) [17], if we can permute the rows and columns in such a way that the ones in every column and in every row occur consecutively, i.e. A binary matrix has the SC1P if it has both the C1P for rows and the
C1P for columns. There exist several linear-time and polynomial-time algorithms for testing the C1P for columns \([4,9,10,12,13,15]\). These algorithms can also be used for testing the C1P for rows. The column permutation (if one exists) to obtain the C1P for rows will not alter the row permutation (if one exists) that achieved the C1P for columns and vice versa. Thus, testing the SC1P can also be done in linear time.

Not all binary matrices have the SC1P. We consider the Simultaneous Consecutive Ones Submatrix (SC1S) and Simultaneous Consecutive Ones Editing problems (SC1E) to deal with matrices that do not have the SC1P. SC1S problems focus on deleting a minimum number of rows and columns to establish the SC1P whereas SC1E problems deal with flipping a minimum number of 1s to obtain the SC1P. We pose the following optimization problems: i) SC1S-row deletion and ii) SC1S-column deletion in the SC1S category and SC1P-1-Flipping in the SC1E category. Given a binary matrix, the SC1S-row deletion finds a minimum number of rows to be deleted such that the resultant matrix has the SC1P. The SC1S-column deletion finds a minimum number of columns to be deleted such that the resultant matrix has the SC1P. The SC1P-1-Flipping finds a minimum number of 1-entries to be flipped such that the resultant matrix has the SC1P. We refer to the parameterized versions of the above problems parameterized by \(k\) as \(k\text{-SC1S-R}\), \(k\text{-SC1S-C}\) and \(k\text{-SC1P-1E}\) respectively, with \(k\) being the number of rows that can be deleted, columns that can be deleted and 1-entries that can be flipped respectively.

A matrix can be considered as a set of rows (columns) together with an order on this set \([4]\). Throughout this paper, the term matrix refers to a binary matrix. Matrix having at most \(x\) ones in each column and at most \(y\) ones in each row is denoted as \((x, y)\)-matrix. A matrix that has at most two ones in each column and at most two ones in each row is denoted by \((2, 2)\)-matrix. A \((2, +)\)-matrix can contain at most two ones per column and no bound on the number of ones per row. A \((+, 2)\)-matrix have no restriction on the number of ones per column and at most two ones per row. Given an \(m \times n\) matrix \(M\), let \(R(M) = \{r_1, r_2, \ldots, r_m\}\) and \(C(M) = \{c_1, c_2, \ldots, c_n\}\) denote the sets of rows and columns, respectively. Here, \(r_i\) and \(c_j\) denote the binary vectors corresponding to row \(r_i\) and column \(c_j\) of \(M\), respectively. The \((i, j)\)-th entry in \(M\) is denoted as \(m_{ij}\). Let \(A = \{ij \mid m_{ij} = 1\}\) be the set of all 1-entries in \(M\). For a subset \(R' \subseteq R(M)\) of rows, \(M[R']\) and \(M \setminus R'\) denote the submatrix induced on \(R'\) and \(R(M) \setminus R'\) respectively \([15]\). Similarly, for a subset \(C' \subseteq C(M)\) of columns, the submatrix induced on \(C'\) and \(C(M) \setminus C'\) are denoted by \(M[C']\) and \(M \setminus C'\), respectively.

We present the formal definitions of the problems \(k\text{-SC1S-R}\) and \(k\text{-SC1S-C}\) as follows.

**\(k\text{-SC1S-R}\)**

**Instance:** \(< M, k >\)- An \(m \times n\) binary matrix \(M\) and an integer \(k \geq 0\).

**Parameter:** \(k\).

**Question:** Does there exist a set of rows \(R' \subseteq R(M)\), such that \(|R'| \leq k\) and \(M \setminus R'\) has the SC1P?

**\(k\text{-SC1S-C}\)**

**Instance:** \(< M, k >\)- An \(m \times n\) binary matrix \(M\) and an integer \(k \geq 0\).

**Parameter:** \(k\).

**Question:** Does there exist a set of columns \(C' \subseteq C(M)\), such that \(|C'| \leq k\) and \(M \setminus C'\) has the SC1P?
Oswald and Reinelt [17] posed the following \textit{k-SC1P-1E} problem as \textit{k-augmented simultaneous consecutive ones property}.

\begin{tabular}{|l|}
\hline
\textbf{Instance:} An \(m \times n\) binary matrix \(M\) and an integer \(k \geq 0\). \\
\textbf{Parameter:} \(k\). \\
\textbf{Question:} Does there exist a set of \(1\)-entries \(A' \subseteq A\), with \(|A'| \leq k\) such that the resultant matrix obtained by flipping the entries of \(A'\) in \(M\) has the \(SC1P\)? \\
\hline
\end{tabular}

**Complexity Status:** Oswald and Reinelt [17] have shown that \textit{k-SC1P-1E} is NP-complete. However, the parameterized complexity of this problem is open. To the best of our knowledge, the parameterized problems posed under the simultaneous consecutive ones submatrix (\textit{SC1S}) category are not explicitly mentioned in the literature and the parameterized complexity of all the above problems on (2, \(\ast\))-matrices and (\(\ast\), 2)-matrices are open prior to this work.

**Our Results:** Our aim is to investigate the fixed-parameter tractability and approximability of the parameterized versions of \textit{SC1S} and \textit{SC1E} problems defined above. We show that the problems \textit{k-SC1S-R} and \textit{k-SC1S-C} are NP-complete. For (2, 2)-matrices, we show that \textit{k-SC1S-R}, \textit{k-SC1S-C} and \textit{k-SC1P-1E} are solvable in polynomial-time. We present FPT algorithms for \textit{k-SC1S-R}, \textit{k-SC1S-C} and \textit{k-SC1P-1E} with run-times \(O^*(4^k)\) and \(O^*(6^k)\) respectively on (2, \(\ast\))-matrices. We also present FPT algorithms for \textit{k-SC1S-R}, \textit{k-SC1S-C} and \textit{k-SC1P-1E} with run-times \(O^*(3^k)\), \(O^*(4^k)\) and \(O^*(6^k)\) respectively on (\(\ast\), 2)-matrices. We show that using our algorithm, \textit{k-SC1S-R}, \textit{k-SC1S-C} and \textit{k-SC1P-1E} can be approximated in polynomial-time with a factor of 4, 3 and 6 respectively on (2, \(\ast\))-matrices and with a factor of 3, 4 and 6 respectively on (\(\ast\), 2)-matrices.

**Applications:** In bioinformatics, to obtain clusters from the metabolic networks [11], the adjacency matrix of metabolites were transformed to a matrix having the \textit{SC1P} by flipping 0’s to 1’s. In the same way, we consider the problem of finding clusters by flipping 1’s to 0’s to establish the \textit{SC1P}, which is defined as follows:

\begin{tabular}{|l|}
\hline
\textbf{Finding Clusters} \\
\textbf{Instance:} An adjacency matrix of metabolites \(M\), and an integer \(k \geq 0\). \\
\textbf{Parameter:} \(k\). \\
\textbf{Question:} Does there exist a set of \(1\)-entries of size atmost \(k\) whose flipping results in matrix with the \textit{SC1P}? \\
\hline
\end{tabular}

The fixed-parameter tractability of \textit{k-SC1P-1E} on (2, \(\ast\))-matrices and (\(\ast\), 2)-matrices shows that finding clusters is also FPT on (2, \(\ast\))-matrices and (\(\ast\), 2)-matrices.

**Techniques Used:** Our results rely on the following forbidden submatrix characterization of the \textit{SC1P} (see Figure [1] by Tucker [19].

\begin{quote}
\textbf{Theorem 1.} ([19, Theorem 11]) A matrix \(M\) has the \textit{SC1P} if and only if no submatrix of \(M\), or of the transpose of \(M\), is a member of the configuration (see Section [2]) of \(M_1\) \((k \geq 1)\), \(M_2\), \(M_{22}\), \(M_{31}\), \(M_{32}\), and \(M_{33}\).
\end{quote}

\footnote{\(O^*\) notation ignores the polynomial terms and focuses on exponential part [6].}
FPT results of Simultaneous Consecutive Ones Problems.

We used the following theorem [17] which is exactly same as the above theorem.

**Theorem 2.** A matrix $M$ has the SC1P if and only if no submatrix of $M$ is a member of the configuration of $M_{k, 1}$, $M_{k, 2}$, $M_{k, 3}$, $M_{2, k}$, $M_{3, k}$, or their transposes.

**Related Work:** In 1972, Tucker [19] characterized matrices having the SC1P using a set of forbidden submatrices. Later Oswald considered the weighted simultaneous consecutive ones problem (WSC1P) (Chapter 4, [16]) and is defined as follows. Given two $m \times n$ matrices $A$ and $B$, and a cost matrix $C = (c_{ij}) \in \mathbb{R}^{(m \times n)}$, the total cost involved in switching entries to transform $B$ to $A$ is given by $c(A) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} |b_{ij} - a_{ij}|$. For a given matrix $B$, WSC1P problem finds a matrix $A$ which has the SC1P and minimizes $c(A)$. The WSC1P was shown to be NP-hard through a related problem namely $k$-augmented simultaneous consecutive ones property ($k$-SC1P-1E). The restricted version of the WSC1P, in which no row-permutations and column-permutations were allowed was also shown to be NP-hard. Oswald [16] also studied polyhedral aspects of the simultaneous consecutive ones polytope.

**Organization of the paper:** In Section 2, we give necessary preliminaries and observations. In Section 3, we present hardness results for the problems $k$-SC1S-R, $k$-SC1S-C and $k$-SC1P-1E and fixed-parameter algorithms for these problems on $(2, +)$ and $(+, 2)$-matrices. Last section draws conclusions and gives an insight to further work.

## 2 Preliminaries

In this section, we present definitions and notations related to binary matrix and graphs associated with binary matrices. All graphs discussed in this paper shall always be undirected and simple. A graph $G$ is defined as a tuple $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of vertices and $E = \{e_1, e_2, \ldots, e_m\}$ is a finite set of edges. Throughout this paper, we consider $|V| = n$ and $|E| = m$ respectively. We refer the reader to [21] for the standard definitions and notations related to graphs. A sequence of distinct vertices $(u_1, u_2, \ldots, u_n)$ with $u_i$ adjacent to $u_{i+1}$ for each $1 \leq i < n$ is called a $u_1 - u_n$ path. A Hamiltonian path is a path...
that visits every vertex exactly once. A cycle is a graph consisting of a path \((u_1u_2\ldots u_n)\) and the additional edge \(\{u_n,u_1\}\). The length of a path (cycle) is the number of edges present in it.

A chord in a cycle is an edge that is not part of the cycle but connects two non-consecutive vertices in the cycle. A hole or chordless cycle is a cycle of length at least four, where no chords exist. A graph is chordal if it contains no hole, i.e. In a chordal graph, every cycle of length at least 4 contains a chord.

**Lemma 3.** (Theorem 2) In a graph \(G = (V,E)\), a chordless cycle can be detected in \(O(|V| + |E|)\)-time, where \(|V|\) and \(|E|\) are the number of vertices and edges in \(G\) respectively.

Two vertices \(u, v\) in \(V\) are connected if there exists a path between \(u\) and \(v\) in \(G\). A graph is connected if there exists a path between every pair of vertices. A graph \(G = (V', E')\) is a subgraph of \(G\) if \(V' \subseteq V\) and \(E' \subseteq E\). The subgraph of \(G\) induced by \(V'\), denoted as \(G[V']\), is the graph \(G' = (V', E')\) with \(V' \subseteq V\) and \(E' = \{\{v, w\} \in E \mid v \in V'\text{ and } w \in V'\}\).

A connected component of \(G\) is a maximal connected subgraph of \(G\). Deletion of a vertex \(v \in V\) means, deleting \(v\) and all edges incident on \(v\). A graph \(G = (V,E)\) is bipartite if \(V\) can be partitioned into two disjoint vertex sets \(V_1\) and \(V_2\) such that every edge in \(E\) has one endpoint in \(V_1\) and the other endpoint in \(V_2\). A bipartite graph is often denoted as \(G = (V_1, V_2, E)\), where \(V_1\) and \(V_2\) are the two partitions of \(V\). A graph is an interval graph if its vertices can be assigned real line intervals such that there is an edge between two vertices if and only if their corresponding intervals intersect.

Given an \(m \times n\) matrix \(M\), the \(n \times m\) matrix \(M'\) with \(m'_{j,i} = m_{i,j}\) is called the transpose of \(M\), and is denoted by \(M^T\). Two matrices \(M\) and \(M'\) are isomorphic, if \(M\) is a permutation of the rows and columns of \(M'\). We say, a matrix \(M\) contains \(M'\), if \(M\) contains a submatrix that is isomorphic to \(M'\). The configuration of an \(m \times n\) matrix \(M\) is defined to be the set of all \(m \times n\) matrices which can be obtained from \(M\) by row and column permutations.

For a matrix \(M\), we define weight (an integer value), to each row, column and entry as follows:

**Definition 4.** 1. The weight of a row (column) is equal to the number of times that row (column) appears in \(M\).

2. The weight of an entry is obtained by subtracting one from the sum of the weight of its row and column.

**Definition 5.** The half adjacency matrix \(\Box\) of a bipartite graph \(G = (V_1, V_2, E)\) with \(V_1 = \{u_1, \ldots , u_{n_1}\}\) and \(V_2 = \{v_1, \ldots , v_{n_2}\}\) is an \(n_1 \times n_2\) matrix \(M\) with \(m_{i,j} = 1\) if and only if \(\{u_i, v_j\} \in E\), where \(1 \leq i \leq n_1\) and \(1 \leq j \leq n_2\).

Every matrix \(M\) can be viewed as the half adjacency matrix of a bipartite graph, referred to as the representing graph \(G_M\) of \(M\). The representing graph \(G_M\) \(\Box\) of a matrix \(M_{m \times n}\) is obtained as follows:

**Definition 6.** For a matrix \(M\), \(G_M\) contains a vertex corresponding to every row and column of \(M\), and there is an edge between two vertices corresponding to \(i\)th row and \(j\)th column of \(M\) if and only if the corresponding entry \(m_{ij} = 1\), where \(1 \leq i \leq m\) and \(1 \leq j \leq n\).

A graph \(G\) can also be represented using edge-vertex incidence matrix denoted by \(M(G)\) and is defined as follows.

**Definition 7.** For a graph \(G = (V,E)\), the rows and columns of \(M(G)_{m \times n}\) correspond to edges and vertices of \(G\) respectively. The entries of \(M(G)\) is defined by \(m_{ij} = 1\), if edge \(e_i\) is incident on vertex \(v_j\), and \(m_{ij} = 0\) otherwise, where \(1 \leq i \leq m\) and \(1 \leq j \leq n\).
FPT results of Simultaneous Consecutive Ones Problems.

We obtain the following Lemma from ([4, Theorem 2.2]).

- **Lemma 8.** If $G$ is a connected graph and the edge-vertex incidence matrix $M(G)$, of $G$ has the C1P for rows, then $G$ is a path.

Matrices having the SC1P are characterized by a forbidden submatrix characterization (Theorem [2]). For ease of reference, the notation $X$ is defined as follows.

- **Definition 9.** The fixed-size forbidden submatrices in the forbidden submatrix characterization of SC1P is denoted by $X$.

We obtain the following Lemma from ([4, Proposition 3.4]), by considering the maximum possible size of the forbidden submatrix in $X$.

- **Lemma 10.** Let $M$ be a matrix of size $m \times n$. Then a minimum size submatrix in $M$ that is isomorphic to one of the forbidden submatrices of $X$ can be found in $O(m^3n)$ time.

If $M$ is a $(2, *)$-matrix or $(*, 2)$-matrix of size $m \times n$. Then a minimum size submatrix in $M$ that is isomorphic to one of the forbidden submatrices of $X$ can be found in $O(m^n)$ time and $O(m^3n)$ time respectively.

We obtain the following Lemma from ([4, Proposition 3.4]), by considering the maximum number of ones in each row of $M$ as $n$.

- **Lemma 11.** Let $M$ be a matrix of size $m \times n$. Then a minimum size submatrix of type $M_{ik}$ or $M_{ik}^T$ $(k \geq 1)$ in $M$ can be found in $O(nm^3)$ time.

If $M$ is a $(2, *)$-matrix or $(*, 2)$-matrix of size $m \times n$. Then a minimum size submatrix of type $M_{ik}$ or $M_{ik}^T$ $(k \geq 1)$ in $M$ can be found in $O(n^2m^3)$ time and $O(m^3)$ time respectively.

To destroy $M_{ik}/M_{ik}^T$, we consider the representing graph $G_{M_{ik}}/G_{M_{ik}}^T$. Following result shows that $G_{M_{ik}}/G_{M_{ik}}^T$ is a chordless cycle.

- **Lemma 12.** ([4, Observation 3.1]) The representing graph $G_{M_{ik}}/G_{M_{ik}}^T$ is a chordless cycle of length $2k + 4$.

## 3 Our Results

In this section, firstly we present the hardness results for SC1S problems. Then we present the polynomial-time solvability of SC1S and SC1E problems on $(2,2)$-matrices. We also present FPT algorithms for SC1S and SC1E problems on $(2,*)$ and $(*,2)$-matrices.

**Preprocessing** on the input matrix $M$ is done as follows: Assign weights (Definition 3) to each row, column, and entry and delete all but one occurrence of identical rows and columns.

The resultant matrix thus obtained will have no identical rows and columns, and it is also possible for a matrix to have more than one row/column/entry with equal weight.

### 3.1 Hardness Results

Even though the number of forbidden submatrices to establish the SC1P is less than the number of forbidden submatrices for the C1P, the problems posed in this paper, to obtain the SC1P turn out to be difficult. Here, we show that the problems $k$-SC1S-R and $k$-SC1S-C are NP-complete. The following theorem proves the NP-completeness of the $k$-SC1S-R problem using Hamiltonian path as a candidate problem.
Theorem 13. Given an \( m \times n \) matrix \( M \), deciding if there exists a set of rows \( R' \subseteq R(M) \), such that \( |R'| \leq k \) and \( M \setminus R' \) has the SC1P is NP-complete.

Proof. We first show that \( k\text{-SC1S-R} \in \text{NP} \). Given a matrix \( M \) and an integer \( k \), the certificate chosen is the given set of rows \( R' \subseteq R(M) \). The verification algorithm affirms that \( |R'| \leq k \), and then it checks whether deleting these \( k \) rows from \( M \) yields a matrix with the SC1P. This certificate can be verified in polynomial-time.

We prove that \( k\text{-SC1S-R} \) problem is NP-hard by showing that Hamiltonian-Path \( \leq_p \) \( k\text{-SC1S-R} \). Let \( G \) be a graph and \( M(G) \) be the edge-vertex incidence matrix (Definition \[7\]) obtained from \( G \). Without loss of generality, assume that \( G \) is connected and let \( k = m - n + 1 \). We show that \( G \) has a Hamiltonian path if and only if there exists a set of rows of size \( k \) in \( M(G) \) whose deletion results in a matrix \( M'(G) \), that satisfy the SC1P.

Assume that \( G \) contains a Hamiltonian path. In \( M(G) \), delete the rows that correspond to edges which are not part of the Hamiltonian path in \( G \). Since Hamiltonian path contains \( n-1 \) edges, the number of rows remaining in \( M(G) \) will be \( m - k \) and hence the number of rows deleted will be \( k \). Now order the columns and rows of \( M(G) \) with respect to the sequence of vertices and edges in the Hamiltonian path respectively. Clearly, the resulting matrix has the SC1P.

To prove the other direction, let \( M'(G) \) be the matrix obtained by deleting \( k \) rows from \( M(G) \) and assume that \( M'(G) \) has the SC1P. Now, the number of rows in \( M'(G) \) is \( m - k \). Let \( G' \) be the subgraph obtained from \( M'(G) \), by considering \( M'(G) \) as an edge-vertex incidence matrix (Definition \[7\]) of \( G' \). Since \( M'(G) \) has the SC1P, it has the C1P for rows. Also, note that \( M'(G) \) has \( n-1 \) rows. This implies that \( G' \) is a path (Lemma \[8\]) of length \( n-1 \), which clearly indicates that \( G \) has a Hamiltonian path. The column permutation needed to convert \( M'(G) \) into a matrix that has strong C1P for rows gives the relative order of vertices of \( G \)'s Hamiltonian path. This proves the NP-completeness of \( k\text{-SC1S-R} \).

Corollary 14. The problem \( k\text{-SC1S-C} \) is NP-complete.

The NP-completeness of \( k\text{-SC1S-C} \) can be proved similar to Theorem 13 (NP-completeness of \( k\text{-SC1S-R} \)) by considering \( M \) as the vertex-edge incidence matrix and \( k \) as the number of columns to be deleted.

3.2 Easily Solvable Instances

The problems \( k\text{-SC1S-R} \), \( k\text{-SC1S-C} \) and \( k\text{-SC1P-1E} \) defined in Section 1 are solvable in polynomial-time on \((2,2)\)-matrices.

Theorem 15. On \((2,2)\)-matrices, \( k\text{-SC1S-R} \) is polynomial-time solvable.

Proof. A \((2,2)\)-matrix can only contain forbidden submatrices \( M_{I_k} \) and \( M_{I_k}^T \) (where \( k \geq 1 \)) of unbounded size. Since a matrix can be viewed as the half adjacency matrix of a bipartite graph, the \( k\text{-SC1S-R} \) problem on such matrices can be formulated as a graph modification problem. Here, modification means, deletion of a vertex/edge or addition of an edge in a graph. Let \( G_M \) be the representing graph (Definition \[3\]) of the preprocessed (Section 3) input matrix \( M \). Since each column and row of \( M \) contains at most two ones, the degree of each vertex in \( G_M \) is at most two. So the connected components of \( G_M \) are chordless cycles or paths. From Lemma \[12\] it is clear that the representing graph of an \( M_{I_k} \) (as well as its transpose \( M_{I_k}^T \), where \( k \geq 1 \)) is a chordless cycle. To destroy \( M_{I_k} \) and \( M_{I_k}^T \), it is sufficient to destroy chordless cycles in \( G_M \). For each cycle \( C \) of \( G_M \), consider the submatrix \( M[C] \) induced by the vertices of \( C \). To destroy \( C \), delete the vertex \( v \) in \( C \), corresponding to a row
FPT results of Simultaneous Consecutive Ones Problems.

(a) Forbidden submatrix $M_3$, (b) $G_{M_3}$, the representing graph of $M_3$, (c) Forbidden submatrix $M_1$, and (d) $G_{M_3}$, the representing graph of $M_3$.

Figure 2 (a) Forbidden submatrix $M_3$, (b) $G_{M_3}$, the representing graph of $M_3$, (c) Forbidden submatrix $M_1$, and (d) $G_{M_3}$, the representing graph of $M_3$.

$r$ having minimum weight (Definition 3) by breaking ties arbitrarily in $M[C]$. Decrement the parameter $k$ by the weight of row $r$ and delete $r$ from $M$. The input is an Yes-instance, if the sum of the weight of all the rows removed from $M$ is at most $k$, otherwise a No-instance.

The representing graph $G_M$ can be constructed from $M$ in polynomial time. Since $G_M$ has only finite number of vertices and degree of each vertex is at most two, $G_M$ contains only finite number of cycles. Using Lemma 3 each chordless cycle can be detected in $O(|V| + |E|)$-time. Therefore for $(2, 2)$-matrices, $k$-SC1S-R can be solved in $O(k(|V| + |E|))$-time.

Corollary 16. For $(2, 2)$-matrices, the problems $k$-SC1S-C and $k$-SC1P-1E are polynomial-time solvable.

Algorithms for solving $k$-SC1S-C and $k$-SC1P-1E problems on $(2, 2)$-matrices are similar to the algorithm for solving $k$-SC1S-R (Theorem 15), except that they differ only in the way the chordless cycles are destroyed. In $k$-SC1S-C, deletion of a column corresponds to a vertex deletion in the corresponding representing graph. In $k$-SC1P-1E, flipping a 1-entry corresponds to an edge deletion in the associated representing graph.

### 3.3 Fixed-Parameter Tractable (FPT) algorithms on restricted matrices

In this section, we present FPT algorithms for the problems defined in Section 1 on $(2, *)$-matrices and $(*, 2)$-matrices. Here, we make use of the forbidden submatrix characterization for the SC1P by Tucker [19]. If $M$ is a $(2, *)$-matrix then the forbidden matrices in $X$ that can be contained in $M$ are $M_{11}, M_{11}^T$ and $M_{31}^T$ because all other matrices in $X$ contain a column with more than two ones whereas if $M$ is a $(*, 2)$-matrix, then the forbidden matrices in $X$ that can be contained in $M$ are $M_{11}, M_{11}^T$ and $M_{31}$, because all other matrices in $X$ contain a row with more than two 1s.

Our algorithm consists of two stages. Given an input matrix $M$, the first stage preprocess (Section 3) the input matrix, and then iteratively searches and destroys every submatrix that contains one of the forbidden submatrices in $X$ (Definition 9). For this, we use a recursive branching algorithm, which is a search tree that branches recursively into several subcases, depending upon the problem under consideration. If the resultant matrix obtained after the first stage does not have the SC1P, then the second stage of our algorithm focuses on destroying the forbidden submatrices of type $M_{1k}$ and $M_{1k}^T$ (where $k \geq 2$) efficiently.

In the second stage of our algorithm, branching strategy cannot be applied to destroy $M_{1k}$ and $M_{1k}^T$ (where $k \geq 2$), because their sizes are unbounded. We use the result of the following theorem cleverly to get rid of $M_{1k}$ and $M_{1k}^T$ in the second stage.

Theorem 17. Let $M$ be a $(2, *)$-matrix or $(*, 2)$-matrix that does not have identical columns and rows. If $M$ does not have the SC1P and does not contain matrices in $X$ as submatrices,
then the matrices of type $M_{k}$ and $M_{k}^T$ (where $k \geq 2$) that are contained in $M$ are pairwise disjoint, i.e. they have no common column or row.

**Proof.** Consider the representing graph $G_M$ (Definition 5) of a given $m \times n$ matrix $M$. Since $M$ is a $(2,\ast)$-matrix or $(\ast,2)$-matrix that does not have identical rows and columns, there are no chordless cycles of length 4 in $G_M$. Assume that $M$ contains a pair of matrices of type $M_{k}$ and/or $M_{k}^T$ (where $k \geq 2$) that share at least one common column or row. This implies that there are two induced cycles of length at least 8 in $G_M$ that have at least one vertex in common corresponding to a column or row of $M$ (Lemma 12). Since these two cycles share a common vertex, $G_M$ contain either a $G_{M_{31}}$ or $G_{M_{31}^T}$ as an induced subgraph. This means that $M$ contains an $M_{31}$ or $M_{31}^T$ and is a contradiction.

### An FPT Algorithm for $k$-SC1S-R

In Algorithm 1 we present an FPT algorithm $k$-SC1S-row-deletion for solving $k$-SC1S-R problem on $(2,\ast)$-matrices and $(\ast,2)$-matrices. Given a matrix $M$ and parameter $k$ (maximum number of rows that can be deleted), Algorithm 1 first preprocess (Section 3) the input matrix, and then searches and destroys every forbidden submatrix that belongs to $X$ (Definition 2). If $M$ contains a forbidden submatrix in $X$, then the algorithm branches into at most four/three subcases (depending on whether $M$ is a $(2,\ast)$/$(\ast,2)$-matrix) each corresponds to deleting a row from the forbidden submatrix found in $M$. In each of the subcases, when a row is deleted, the parameter $k$ is decremented by the weight (Definition 1) of that row. As long as $k > 0$, the above steps are repeated for each subcase until all the forbidden submatrices of $X$ are destroyed. If the resulting matrix still does not have the SC1P, then the only possible forbidden submatrices that can remain in $M$ are of type $M_{31}$ and $M_{31}^T$ (where $k \geq 2$). If they appear in $M$, by Theorem 17 they are pairwise disjoint. Pairwise disjoint $M_{k}$ and $M_{k}^T$ in $M$, can be destroyed by deleting a row with minimum weight (by breaking ties arbitrarily) from each of them. On deletion of a row, the parameter $k$ is decremented by the weight of that row. Using Lemma 11 a minimum size forbidden submatrix of type $M_{k}$ and $M_{k}^T$ can be found in polynomial-time. The number of pairwise disjoint $M_{k}$ and $M_{k}^T$ in $M$ is $O(\min(m,n))$ (the smallest forbidden submatrix of type $M_{k}$ and $M_{k}^T$ has size $4 \times 4$). Therefore all submatrices of type $M_{k}$ and $M_{k}^T$ in $M$ can be destroyed in polynomial-time and $k$-SC1S-row-deletion algorithm runs in $O^*(4^k)/O^*(3^k)$-time on $(2,\ast)/((\ast,2)$-matrices.

The correctness of the branching step is explained in the following Lemma.

**Lemma 18.** Let $M$ be a $(2,\ast)$-matrix that does not have the SC1P. Suppose $M$ contains one of the forbidden submatrices in $X$. Let $M[[r_1,r_2,r_3,r_4]]$ be a submatrix that contains a forbidden submatrix in $X$ where \{r_1,r_2,r_3,r_4\} \subseteq R(M).$ Then, any solution of $k$-SC1S-R includes at least one of the rows $r_1,r_2,r_3,r_4$.

**Proof.** Assume that there exists a solution for $k$-SC1S-R, say $S$ that contains none of the rows $r_1,r_2,r_3,r_4$. Let $M' = M \setminus S$ be the matrix with the SC1P. This implies that $M'[[r_1,r_2,r_3,r_4]]$ in $M'$ satisfies the SC1P, which is a contradiction.

Algorithm 1 can be used to solve $k$-SC1S-R on $(2,\ast)$-matrices by considering the number of branches as three (since the largest forbidden matrix in $X$ that can occur in a $(2,\ast)$-matrix $M$ has 3 rows). The correctness of the branching step can be explained similar to that of Lemma 18.

**Theorem 19.** $k$-SC1S-R on a $(2,\ast)$-matrix/$(\ast,2)$-matrix $M_{m \times n}$, can be solved in $O^*(4^k)$-time/$O^*(3^k)$-time, where $k$ denotes the number of rows that can be deleted. Consequently it is FPT.
Algorithm 1 Algorithm \( k\text{-SC1S-row-deletion}(M, k) \)

**Input:** An instance \( \langle M_{m \times n}, k \rangle \) where \( M \) is a \((2, *)\)-binary matrix and \( k \geq 0 \)

**Output:** Return Yes, if there exists a set of rows \( R' \subseteq R(M) \), with \( |R'| \leq k \) such that \( M \setminus \{ R' \} \) has the SC1P, otherwise return No.

**Stage 1:**
1. Apply preprocessing (Section 3) in \( M \).
2. if \( M \) has the SC1P and \( k \geq 0 \) then return Yes.
3. if \( k < 0 \) then return No.

**Branching Step:**
4. if there exists a submatrix \( M' \) in \( M \) that is isomorphic to one of the forbidden submatrices in \( \{ M_{11}, M_{11}^T, M_{22}^T \} \) then,
   Branch into at most four instances \( I_i = \langle M_i, k_i \rangle \) where \( i \in \{ 1, 2, 3, 4 \} \)
   Set \( M_i \leftarrow M \setminus \{ r_i \} \) // \( r_i \) denotes the \( i \)th row of \( M' \).
   Set \( k_i \leftarrow k - 1 \) // Decrement parameter by 1.
   For some \( i \in \{ 1, 2, 3, 4 \} \), if \( k\text{-SC1S-row-deletion}(M_i, k_i) \) returns Yes, then return Yes, else if all instances return No, then return No.

**Stage 2:**
For each of the leaf instances \( M_i \) where \( i \in \{ 1, 2, \ldots, 4^k \} \), perform the following.
5. while there exists a submatrix \( N \) in \( M_i \) that is isomorphic to an \( M_{lk} \) or \( M_{lk}^T \) and \( k_i > 0 \) do
6. Delete a row \( r \) with minimum weight in \( N \) from \( M_i \).
7. Decrement the parameter \( k_i \) by the weight of the deleted row \( r \).
8. if \( M' \) does not contain \( M_{lk} \) or \( M_{lk}^T \) and \( k_i \geq 0 \) then
9. return Yes
10. else
11. return No

If any of the above leaf instances returns Yes, then return Yes, else return No.

**Proof.** Algorithm \( k\text{-SC1S-row-deletion} \) explained in Algorithm 1 employs a search tree. Each node in the search tree has at most four/three subproblems, and therefore the tree has at most \( 4^k/3^k \) leaves. The size of the search tree is \( O(4^k)/O(3^k) \). A submatrix \( M' \) of \( M \) that is isomorphic to one of the forbidden matrices in \( X \) can be found in \( O(m^4n) \)-time (using Lemma 11). Therefore the time required for destroying forbidden submatrices in \( X \) from \( M \) (stage 1) is \( O(4^km^4n)/O(3^km^3n) \). The time required for finding a submatrix of type \( M_{lk} \) and \( M_{lk}^T \), (where \( k \geq 2 \)) from \( M \) is \( O(m^3n^3)/O(m^3) \) (using Lemma 11). For each of the leaf instance \( M_i \), line 4 of Algorithm 1 is executed at most \( k_i \) times and \( k_i \leq k \). Therefore the time complexity of destroying \( M_{lk} \) and \( M_{lk}^T \) from \( M \) (stage 2) is \( O(4^km^3n^3k)/O(3^km^3k) \). The total time complexity of Algorithm 1 is \( O(4^k(m^4n + k.m^3n^3))/O(3^k(m^3n + k.m^3)) \). ▶

**Corollary 20.** \( k\text{-SC1S-R} \) on a \((2, *)\)-matrix/\((*, 2)\)-matrix \( M_{m \times n} \) can be approximated in polynomial-time with a factor of four/three.

For the approximation scenario, in Stage 1 of Algorithm 1 instead of branching on each of the rows of a forbidden submatrix of \( X \) found in \( M \), delete all rows of each of the forbidden submatrix in \( X \). From Algorithm 1 it is clear that Stage 2 solves the problem exactly. This results in a 4-factor/3-factor approximation algorithm.

A related problem of deleting minimum number of columns to get the SC1P \((k\text{-SC1S-C problem})\) can also be solved using Algorithm 1 (consider the columns instead of rows in...
An FPT Algorithm for $k$-SC1P-1E

The $k$-SC1P-1E problem can also be solved using Algorithm 1 with a modification in the branching step as follows. Here we branch on the number of 1-entries of a forbidden submatrix of $X$ found in $M$. In each branch, we flip the corresponding 1-entry and the parameter $k$ is decremented by the weight of that 1-entry (Definition 4). The number of 1-entries in the largest forbidden submatrix of $X$ is 6 (for both $(2,\ast)$-matrix and $(\ast,2)$-matrix), which leads to a branching factor of at most 6. After the branching step, the remaining pairwise disjoint forbidden submatrices of type $M_Ik$ and $M_Ik^T$ (where $k \geq 2$) in $M$ can be destroyed in polynomial time by flipping a minimum weight 1-entry (Definition 4) in $M_Ik$ and $M_Ik^T$ respectively. After destroying each $M_Ik$ and $M_Ik^T$, branching step is applied to destroy the forbidden submatrices in $X$ that might be created, while flipping 1-entries in $M_Ik/M_Ik^T$. Therefore the total time complexity is $O^*(6^k)$, which leads to the following theorem.

Theorem 21. $k$-SC1P-1E on a $(2,\ast)$-matrix or $(\ast,2)$-matrix $M_{m \times n}$ can be solved in $O^*(6^k)$-time where $k$ denotes the number of allowed flippings. $k$-SC1P-1E can be approximated in polynomial-time with a factor of 6.

4 Conclusion

We developed fixed-parameter tractable and polynomial-time approximation algorithms for $k$-SC1P-1E problem on $(2,\ast)$-matrices and $(\ast,2)$-matrices, which was posed as an open problem in [17]. We also show that the SC1S problems, $k$-SC1S-R and $k$-SC1S-C are NP-complete, fixed-parameter tractable and polynomial-time approximable on $(\ast,2)$-matrices and $(2,\ast)$-matrices. We also pose the following problems related to the SC1P: Does there exist structural characterization(s) for matrices with the SC1P, similar to the characterizations for matrices with the C1P given by [7] [12] [14]? Can interval deletion results/techniques in [2] [3] be applied to $k$-SC1S-R, $k$-SC1S-C and $k$-SC1P-1E problems to obtain FPT algorithms on matrices with no bound on the number of 1s in rows and columns?

References

1. Kellogg S Booth and George S Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1
2. Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1096–1115. Society for Industrial and Applied Mathematics, 2016. doi:10.1137/1.9781611974331.ch77
3. Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms (TALG)*, 11(3):21, 2015. doi:10.1137/1.9781611973402.9
4. Michael Dom. Recognition, Generation, and Application of Binary Matrices with the Consecutive Ones Property. Cuvillier, 2009. URL: http://fpt.akt.tu-berlin.de/publications/theses/dom08.pdf
5. Michael Dom, Jiong Guo, and Rolf Niedermeier. Approximation and fixed-parameter algorithms for consecutive ones submatrix problems. *Journal of Computer and System Sciences*, 76(3-4):204–221, 2010. doi:10.1016/j.jcss.2009.07.001
6. Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013. doi:10.1007/978-1-4471-5559-1
7. Delbert Fulkerson and Oliver Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics*, 15(3):835–855, 1965. doi:10.2140/pjm.1965.15.835
8 Michael R Garey and David S Johnson. Computers and Intractability: A guide to the theory of np-completeness. *WH Freemann, New York*, 70, 1979. [doi:10.1137/1024022]

9 Wen-Lian Hsu. A simple test for the consecutive ones property. *Journal of Algorithms*, 43(1):1–16, 2002. [doi:10.1006/jagm.2001.1205]

10 Wen-Lian Hsu and Ross M McConnell. Pct trees and circular-ones arrangements. *Theoretical computer science*, 296(1):99–116, 2003. [doi:10.1016/S0304-3975(02)00435-8].

11 Rainer König, Gunnar Schramm, Marcus Oswald, Hanna Seitz, Sebastian Sager, Marc Zapatka, Gerhard Reinelt, and Roland Eils. Discovering functional gene expression patterns in the metabolic network of escherichia coli with wavelets transforms. *BMC bioinformatics*, 7(1):119, 2006. [doi:10.1186/1471-2105-7-119].

12 Ross M McConnell. A certifying algorithm for the consecutive-ones property. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 768–777. Society for Industrial and Applied Mathematics, 2004. URL: [http://dl.acm.org/citation.cfm?id=982792.982909].

13 João Meidanis, Oscar Porto, and Guilherme P Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88(1-3):325–354, 1998. [doi:10.1016/S0166-218X(98)00078-X].

14 NS Narayanaswamy and R Subashini. A new characterization of matrices with the consecutive ones property. *Discrete Applied Mathematics*, 157(18):3721–3727, 2009. [doi:10.1016/j.dam.2009.08.001].

15 NS Narayanaswamy and R Subashini. Obtaining matrices with the consecutive ones property by row deletions. *Algorithmica*, 71(3):758–773, 2015. [doi:10.1007/s00453-014-9925-1].

16 Marcus Oswald. *Weighted consecutive ones problems*. PhD thesis, 2003. URL: [http://archiv.ub.uni-heidelberg.de/volltextserver/3588/1/diss.pdf].

17 Marcus Oswald and Gerhard Reinelt. The simultaneous consecutive ones problem. *Theoretical Computer Science*, 410(21-23):1986–1992, 2009. [doi:10.1016/j.tcs.2008.12.039].

18 Mathieu Raffinot. Consecutive ones property testing: cut or swap. In *Conference on Computability in Europe*, pages 239–249. Springer, 2011. [doi:10.1007/978-3-642-21875-0_25].

19 Alan Tucker. A structure theorem for the consecutive 1’s property. *Journal of Combinatorial Theory, Series B*, 12(2):153–162, 1972. [doi:10.1016/0095-8956(72)90019-6].

20 Takeaki Uno and Hiroko Satoh. An efficient algorithm for enumerating chordless cycles and chordless paths. In *International Conference on Discovery Science*, pages 313–324. Springer, 2014. URL: [http://arxiv.org/abs/1404.7610].

21 Douglas B. West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2009. URL: [https://ia801204.us.archive.org/35/items/igt_west/igt_west_text.pdf].