Approximation Algorithms for the Consecutive Ones Submatrix Problem on Sparse Matrices ⋆

Jinsong Tan and Louxin Zhang

Department of Mathematics, National University of Singaproe
2 Science Drive 2, Singapore 117543
{mattjs, matzlx}@nus.edu.sg

Abstract. A 0-1 matrix has the Consecutive Ones Property (C1P) if there is a permutation of its columns that leaves the 1's consecutive in each row. The Consecutive Ones Submatrix (COS) problem is, given a 0-1 matrix A, to find the largest number of columns of A that form a submatrix with the C1P property. Such a problem has potential applications in physical mapping with hybridization data. This paper proves that the COS problem remains NP-hard for i) (2, 3)-matrices with at most two 1's in each column and at most three 1's in each row and for ii) (3, 2)-matrices with at most three 1's in each column and at most two 1's in each row. This solves an open problem posed in a recent paper of Hajiaghayi and Ganjali [12]. We further prove that the COS problem is 0.8-approximatable for (2, 3)-matrices and 0.5-approximatable for the matrices in which each column contains at most two 1's and for $(3, 2)$-matrices.

1 Introduction

A 0-1 matrix is said to have the Consecutive Ones Property (C1P) if there is a permutation of its columns that leaves the 1's consecutive in each row. This property was first mentioned by an archaeologist named Petrie in 1899 [15]. In the last three decades, the study of the C1P property for a given 0-1 matrix found different applications in graph theory [7, 15], computer science [5, 9], and genome sequencing [20].

For sequencing large genomes, biologists first construct a clone library consisting of thousands of clones. Each clone represents a DNA fragment of small size. Then, they map each clone and assemble these maps to determine the map of the entire genome using overlapping between the clones (overlapping can be achieved by using a few restriction enzymes). One approach to determine whether two clones overlap or not is based on hybridization with short probes. In this approach, a clone is exposed to a number of probes and which of these probes hybridize to the clone is determined. Therefore, for the map assembly problem with m clones and n probes, the experimental data is a $n \times m$ matrix $A = (a_{ij})$,

⋆ The research was partially supported by the Biomedical Research Council of Singapore (BMRC01/1/21/19/140).

where $a_{ij} = 1$ if clone c_j hybridizes with probe p_i, and $a_{ij} = 0$ otherwise. If the probes are STS's, each probe is unlikely to occur twice in the genome and hence this leads to physical mapping with unique probes [6]. If an experimental data matrix A is error-free, it has the C1P property and there are efficient algorithms for constructing a correct ordering of clones and probes [7, 3, 2]. However, in reality, physical mapping is prone to various errors and the physical mapping problem with error and uncertainties becomes extremely difficult [2, 14, 10, 19]. This motivates us to study the following problem

Consecutive Ones Submatrix (COS)
Instance: A 0-1 matrix A.
Question: Find the largest number of columns of A that form a submatrix with C1P property.

The decision version of this problem is one of the earliest NP-complete problems appearing in the Garey and Johnson's book [8]. However, the NP-completeness proof was misreferred to in the book. Recently, Hajiaghayi and Ganjali gave a proof [12]. Actually, they proved that the COS problem is NP-complete even for 0-1 matrices in which there are at most two 1's in each column and at most four 1's in each row. On the other hand, the COS problem can be solved in polynomial time for 0-1 matrices with at most two 1's in each row and column. Therefore, their work raises the problem of weather the COS problem remains NP-complete or not for i) (3, 2)-matrices which have at most two 1's in each row and at most three 1's in each column and for ii) (2, 3)-matrices which have at most three 1's in each row and at most two 1's in each column.

Studying the COS problem for matrices with a small number of 1's in column and/or rows is not only theoretically interesting, but also practically important. Actually, this sparsity restriction was proposed by Lander and Istrail [citation here ?] with hopes of making the mapping problem tractable. This paper is divided into five sections. We answer the open problem by proving its NP-completeness for the two special cases just mentioned (in Section 2). Furthermore, we prove that the COS problem is 0.8-approximatable for (2, 3)-matrices, 0.5-approximatable for the matrices in which each column contains at most two 1's (in Section 3) and for $(3, 2)$-matrices (in Section 4). Finally, we conclude this work with several remarks in Section 5. Two closely related works are that the different versions of physical mapping with errors are showed to be NP-hard for sparse matrices in [1, 21].

For basic concepts and knowledge in NP-hardness, polynomial time approximation, we refer readers to the book [8] by Garey and Johnson.

2 NP-hardness of the Problems

In this section, we study the decision version of the COS problem: given a 0-1 matrix A and a positive integer K, are there K columns of A that form a submatrix with the C1P property. A 0-1 matrix is a *(2, 3)-matrix* if it has at most two 1's in each column and at most three 1's in each row; similarly, it is a

(3, 2)-matrix if it has at most three 1's in each column and at most two 1's in each row.

2.1 COS for (2, 3)-Matrices is NP-Complete

To prove the COS problem NP-complete for (2, 3)-matrices, we first define a special spanning tree problem and prove it NP-complete. Formally, it is defined as follows.

Definition 1. *A tree is caterpillar if each node of degree $c \geq 3$ is adjacent to at least $c - 2$ leaf nodes.*

Spanning Caterpillar Tree in Degree-3 Graph
Instance: A graph $G = (V, E)$ in which each node has degree at most 3.
Question: Does G contain a spanning caterpillar subtree T?

Lemma 1. *The Spanning Caterpillar Tree in Degree-3 Graph problem is NP-complete.*

Proof. The proof is by a reduction from the Hamiltonian Path problem for cubic graphs in which every node has degree 3 (see the comment in GT39 in [8]).

Given a cubic graph $G = (V, E)$, we construct a new graph $G' = (V', E')$ by inserting a new node in each edge of G. Formally, we have

$$V' = V \cup \{x_{uv} \mid (u, v) \in E\}$$

$$E' = \{(u, x_{uv}), (x_{uv}, v) \mid (u, v) \in E\},$$

where x_{uv} is called an *edge node*. The reduction follows from the fact that there exists a Hamiltonian path in G if and only if there exists a spanning caterpillar subtree of G'. We conclude the proof by proving this fact as follows.

The 'only if' direction is easily seen. If there is a Hamiltonian path P in the cubic G, we obtain a desired spanning caterpillar subtree of G' by replacing each edge (u, v) by a two-edge path $ux_{uv}v$ if (u, v) is in the path P and attaching the inserted node w_{uv} to u otherwise.

For the 'if' direction, we assume that such a spanning caterpillar subtree T exists in G'. By construction of G', any degree-3 node in T must be a degree-3 node in G' and hence corresponds to a node in G. This also implies that each leaf adjacent to a degree-3 node in T must be an edge node. Therefore, by removing the leaves adjacent to degree-3 nodes from T, we obtain a path P'. Obviously, by construction, P' corresponds to a Hamiltonian path in G. □

Next, we proceed to show the NP-completeness of the COS problem for (2, 3)-matrices by a reduction from the Spanning Caterpillar Tree in Degree-3 Graph problem.

Theorem 1. *The decision version of the COS problem is NP-complete for (2, 3)-matrices.*

Proof. The proof is just a refinement of the reduction used to prove the NP-completeness of the COS problem in [12]. Given a $G = (V, E)$ with n nodes and e edges in which each node has degree at most 3, we consider its incidence matrix $B(G) = (b_{ij})$. Recall that $B(G)$ has a row for each node $v_i \in G$ and a column for each edge $e_j \in G$; and b_{ij} is 1 if v_i is incident to e_j in G and 0 otherwise. Since each node in G has degree at most 3, the incidence matrix $B(G)$ has exact two 1's in each column and at most three 1's in each row.

It can be easily seen that a subset of columns corresponds to a subgraph induced by the corresponding edges in G. Moreover, we claim that G has a spanning caterpillar subtree if and only if $B(G)$ has a submatrix of size $n \times (n-1)$ that has C1P property. We conclude the proof by proving this claim.

We start with the 'only if' direction. Suppose G has a spanning caterpillar subtree T in which each degree-3 node is adjacent to at least one leaf. Since G has n nodes, T contains $n-1$ edges. Moreover, the submatrix induced by the columns corresponding to these $n-1$ edges satisfies C1P property. To prove this fact, we consider two cases.

Case 1. If T is just a Hamiltonian path $u_1, u_2, u_3, \cdots, u_n$ of G, then, the columns corresponding to edges $(u_1, u_2), (u_2, u_3), \cdots, (u_{n-1}, u_n)$ (in this order) form an $n \times (n-1)$ submatrix in which all the 1's are consecutive in each row.

Case 2. If T contains degree-3 nodes, then, we consider the a longest path in T. Assume such a longest path P is $u_1, u_2, u_3, \cdots u_m$, where $m < n$. Since G has n nodes and T is a spanning caterpillar subtree, P contains exactly $n - m$ nodes $u_{i_1}, u_{i_2}, \cdots, u_{i_{n-m}}$ of degree 3 in T such that each u_{i_j} is adjacent to a unique leaf v_{i_j} that is not in the path P. By arranging columns corresponding to edges $(u_1, u_2), (u_2, u_3), \cdots, (u_{m_1}, u_m)$ in the same order and inserting the column corresponding to (u_{i_j}, v_{i_j}) between those corresponding to (u_{i_j-1}, u_{i_j}) and (u_{i_j}, u_{i_j+1}) for each j, we obtain an $n \times (n-1)$ submatrix in which all the 1's are consecutive in each row. This is because the row corresponding to a leaf contains only one 1.

Now we show the 'if' direction. Suppose there exists a submatrix of size $n \times (n-1)$ that has C1P property in $B(G)$. The subgraph induced by the corresponding $n-1$ edges must not admit a cycle (otherwise, at least one of the n rows cannot satisfy the consecutive ones property). Hence, the $(n-1)$ edges induce a spanning subtree of G since these edges are incident to n nodes. Suppose T does not satisfy our special requirement that each degree-3 node in it is adjacent to a leaf node. Then, there exists a node $v \in T$ such that v is adjacent to three nodes $v_1, v_2, v_3 \in T$ with $degree(v_i) \geq 2$, $i = 1, 2, 3$. It's easy to see that there is no way to arrange the columns corresponding to $(v_1, v), (v_2, v)$ and (v_3, v) in $B(G)$ such that all the four rows corresponding to v, v_1, v_2 and v_3 satisfy C1P property. Therefore, the spanning subtree T is caterpillar, i.e, each degree-3 node in it is adjacent to at least a leaf. □

2.2 COS for (3, 2)-Matrices is NP-Complete

In this subsection, we prove the NP-completeness of the COS problem for (3, 2)-matrices by a reduction from the following NP-complete problem for cubic graphs.

Induced Disjoint-Path-Union Subgraph
Instance: Graph $G = (V, E)$ in which each node has degree 3, and a positive integer $k \leq |V|$.
Question: Does there exist a k-node subset $V' \subseteq V$ that induces a subgraph $G(V')$ (not necessarily connected) whose components are paths?

Lemma 2. *The Induced Disjoint-Path-Union Subgraph problem is NP-complete for cubic graphs.*

Proof. According to the theorems by Yannakakis [22] and Lewis [16] (see also problem GT21 and the related comment on page 195 in the book [8] by Garey and Johnson), the problem of finding an induced subgraph with property Π of a particular given size in a cubic graph is NP-complete if Π is:

1) *Nontrivial.* Π is true for a single node and not satisfied by all the graphs in the given domain;

2) *Interesting.* There are arbitrarily large graphs satisfying Π;

3) *Easy.* For a given graph, the property can be verified in polynomial time;

4) *Hereditary.* If a graph satisfies property Π, then any of its induced subgraph must also satisfy property Π.

We now prove our problem is NP-complete by showing that our property π of being a disjoint union of paths satisfies the above conditions.

Let $\mathcal{G}$ denote the set of all cubic graphs. The set $\mathcal{G}'$ of graphs that are isomorphic to an induced subgraph of G is the domain of our property π.

It is easy to see that the property π holds for a single node and an arbitrarily large graph in the domain, but not all graphs in the domain; also, π is hereditary since whenever G is a disjoint union of paths, so is its any subgraph. In addition, the property π can be easily verified in polynomial time. Hence, our problem is NP-complete. □

Theorem 2. *The decision version of the COS problem is NP-complete for (3, 2)-matrices.*

Proof. We prove this NP-completeness result by giving a simple reduction from the Induced Disjoint-Path-Union Subgraph problem for cubic graphs.

Given a cubic graph G with n nodes and m edges, we consider the transpose $B'(G) = (b_{ij})$ of the incidence matrix of G. Thus, $B'(G)$ is a $m \times n$ matrix in which each row corresponds to an edge of G and each column a node of G. The entry b_{ij} is 1 if edge e_i has v_j as an end node; it is 0 otherwise. Since G is a cubic graph, $B'(G)$ has exactly three 1's in each column and two 1's in each row.

We claim that a k-node subset V' induces a subgraph $G(V')$ of G that is a disjoint union of paths if and only if the k columns corresponding to nodes in V' form an $m \times k$ submatrix with C1P property.

The 'only if' direction is similar to Case 1 in the 'only if' condition proof in Theorem 1, so we focus on the 'if' direction. Suppose $B'(G)$ contains an $m \times k$ submatrix C with C1P property. Let V' be the subset of nodes corresponding to the k columns in C. We show that V' induces a subgraph with the desired property. Let v be a node in V'. Since G is cubic, we assume the neighbors of v are v_1 and v_2 and v_3 in G. If all its neighbors v_1, v_2 and v_3 are also in V', then, there is no way to arrange these four columns corresponding to v, v_1, v_2 and v_3 so that the two 1's are consecutive in rows corresponding to (v_1, v) and (v_2, v) and (v_3, v). (Note that each column in the submatrix C contains exactly three 1's.) Hence, at most two of v_i's are in V'. This implies that V' induces a subgraph in which each node has degree at most 2. Furthermore, it is easy to see that the induced subgraph $G(V')$ does not contain a cycle (otherwise, at least one of the rows corresponding to the edges in the cycle cannot have its 1's consecutive under any column permutation). This concludes the proof. □

3 Approximation Algorithms for (2, Δ)-Matrices

We present a 0.8-approximation algorithm for the COS problem for (2, 3)-matrices: Given a (2, 3)-matrix A, find a largest submatrix B of A consisting of a subset of A's columns with the C1P property. We also show that a direct generalization of the algorithm turns out to have an approximation ratio 0.5 for (2, Δ)-matrices. Without loss of generality, we may assume a (2, Δ)-matrix $A = (a_{ij})$ satisfies the following properties in this section:

1. Row-distinguishability. No two rows are identical.
2. Column-distinguishability. No two columns are identical.
3. Connectedness. For any partition of the rows into non-empty subsets R' and R'', there are $i' \in R'$ and $i'' \in R''$ such that $a_{i'j} = a_{i''j} = 1$ for some column j.

Recall that a (2, Δ)-matrix A has at most Δ 1's in each row and at most two 1's in each column. If A contains any column with only one 1, we expand A into a 'full' (2, Δ)-matrix A' whose columns contains exactly two 1's as follows. For each column j containing only one 1, we add an extra row i' that has 1 at the column j and 0 elsewhere. Assume A has k columns with a single 1. Then, its expansion A' has the same number of columns as A and k more rows than A. Finally, we assume A has n columns. For any subset $C \subseteq \{1, 2, \cdots, n\}$, we use $A(C)$ to denote the submatrix of A consisting of columns with indices in C. Then, we give the following simple observation without proof.

Proposition 1. *For any subset $C \subseteq \{1, 2, \cdots, n\}$, $A(C)$ has the C1P property if and only if $A'(C)$ has the property.*

By Proposition 1, the COS problem has the same solution to A and A', and a good approximation solution to A' is also a good approximation to A. Therefore, assume that A has exactly two 1's in each column. Obviously, such a (2, Δ)-matrix A defines uniquely a graph $G(A) = (V, E)$ that has A as the incidence matrix: Each row and column of A corresponds to a node and edge in $G(A)$, respectively. We assume $E = \{1, 2, \cdots, n\}$. For a subset $C \subseteq E$, the subgraph of $G(A)$ induced by C has node set V and edge set C and is denoted by $G_C(A)$.

Proposition 2. *Let $C \subseteq \{1, 2, \cdots, n\}$. Then, $A(C)$ has the C1P property if and only if $G_C(A) = (V, C)$ is a union of caterpillar subtrees. Here a single node is also considered as a caterpillar tree.*

Proof. (Sketch of Proof) For convenience, we use the term edges and nodes in $G(A)$ and columns and rows in A interchangeably. Suppose $G_C(A)$ is a union of caterpillar subtrees C_1, C_2, ..., C_h. For each $1 \leq t \leq h$, using the same discussion as in the proof of Theorem 1, the edges in $C_t = (V_t, E_t)$ form a submatrix $A_{V_t \times E_t}(C)$ with the C1P property. By arranging the columns in each caterpillar subtree in a block and arranging the columns within each block according to their connection in the corresponding subtree, we obtain a submatrix in which all the 1's in each row are arranged consecutively. This is because $A(i, j) = 0$ if node i and edge j don't belong to the same caterpillar tree. Hence, $A(C)$ has the C1P property.

Conversely, suppose $A(C)$ has the C1P property. Using the same discussion as in the proof of Theorem 1, each component of $G_C(A)$ must be a caterpillar subtree. Therefore, $G_C(A)$ is a union of caterpillar trees. □

3.1 A 0.8-approximation Algorithm for (2, 3)-Matrices

Theorem 3. *For the COS problem for any (2, 3)-matrix A, there is a polynomial time 0.8-approximation algorithm.*

Proof. Let the (2, 3)-matrix A have m rows and n columns. Then, $G(A) = (V, E)$ has m nodes and n edges. Since each row contains at most three 1's in A, each node has degree at most 3.

For any subset $C \subseteq \{1, 2, \cdots, n\}$, by Proposition 2, if $A(C)$ has the C1P property, $G_C(A)$ is a union of caterpillar subtrees. Therefore, $G_C(A)$ has at most $m - 1$ edges, and hence C contains at most $(m - 1)$ columns.

To find a 0.8-approximating solution to the COS problem for A, we first find a spanning tree T in $G(A)$. Then, we apply Algorithm A given below to T to find a union of caterpillar subtrees that have at least $0.8(m - 1)$ edges. The set of edges in the union gives a desired solution by Proposition 2. To describe Algorithm A, we recall some basic concepts in graph theory. In a rooted tree, the *depth* of a node is equal to the distance between the root and itself. A non-leaf node is said to be *internal*. For an internal node x in a rooted tree T', we use $T'(x)$ to denote the subtrees rooted at x and $p(x)$ to denote the *parent* of x that is the first node on the unique path from x to the root. Finally, an internal node is said to be *complete* if it is adjacent to 3 internal nodes. Obviously, a complete internal node is of degree 3.

```
ALGORITHM A

Input:  A tree $T$ with $m$ nodes, in which each node has degree at most 3;
Output: A union of caterpillar subtrees with at least $0.8(m-1)$ edges.

1   Pick a leaf $r$ of $T$ and root $T$ at $r$. Let $T_r$ be this rooted tree;
2   Initially, set $RT = T_r$, $RE = \phi$;
3   Do a Breath First Search on $T_r$, for each node $v$ encountered;
4       Record its depth in $T_r$;
5       If $v$ is complete in $T_r$, push it to the end of list $L_c$
6   Repeat the following action until $RT$ contains at most 6 nodes:
7       Pop a node $x$ from the end of $L_c$;
8       if $x$ is complete in $RT$
9           Remove the edge between $x$ and its parent $p(x)$;
10          $RT = RT - RT(x) - (x, p(x))$, $RE = RE \cup \{(x, p(x)\}$;
11  Output $T - RE$;
```

It is easy to see that the algorithm takes $O(m)$ time. Notice that a tree with at most 6 nodes is caterpillar if each node has degree at most 3. Now we prove its connectedness. Now consider a complete internal node x with a largest depth in the tree RT (Note each node *poped* from list L_c at line 7 is with a largest depth in RT). First, the subtree rooted at x, $RT(x)$, is a caterpillar subtree since by assumption every internal node of it is adjacent to at most two internal nodes. Second, since x is a complete internal node, there are at least two internal nodes in $RT(x)$ and hence $RT(x)$ contains at least 4 edges. This implies that each repeat of line 6-10 removes at least 5 edges (including $(x, p(x))$). Therefore, line 6-10 of the algorithm can be repeated at most $(m-1)/5$ times and at most $0.2(m-1)$ edges are removed from the input tree. □

3.2 0.5-Approximation Algorithm for (2, Δ)-Matrices

Now we show that a direct generalization of ALGORITHM A has approximation ratio 0.5 for $(2, \Delta)$-matrices.

Let A be a (2, Δ)-matrix with m rows and n-columns. Then, the corresponding graph $G(A)$ has m nodes and n edges, and each node of it has degree at most Δ. We propose the following generalization to ALGORITHM A.

First, we find a spanning tree T of $G(A)$ and root it at a leaf r. Let the resulting rooted tree be T_r. Obviously, each node in T_r has degree at most Δ. Recall that a non-leaf node is called an internal node. For an internal node x, we use l_x to denote the number of leaves adjacent to it and d_x to denote its degree; x is said to be *complete* if it is adjacent to at least three internal nodes, i.e, $d_x - l_x \geq 3$. We find a union of caterpillar subtrees of T (hence $G(A)$) by repeating the following action until no complete internal nodes exist in T_r:

> Pick a complete internal node x in T_r with the largest depth and then remove the edge $(x, p(x))$ and any other $d_x - l_x - 3$ edges between x and its internal neighbors.

In each repeat, we take away at least $l_x + 1 + 2(d_x - l_x - 1) = 2d_x - l_x - 1$ edges from T_r by removing $d_x - l_x - 2$ edges. Thus, at least half of the edges in T remain in the output union of caterpillar subtrees. Hence we proved that

Theorem 4. *There is a polynomial time 0.5-approximation algorithm for the COS problem when the input matrix has at most two 1's in each column.*

4 A 0.5-Approximation Algorithm for (3, 2)-Matrices

In this section, we use a partition theorem of Lovász in graph theory to obtain a 0.5-approximation algorithm for the COS problem for (3, 2)-matrices. The idea is also generalized to (Δ, 2)-matrices.

4.1 The Algorithm

Recall that a (3, 2)-matrix contains at least three 1's in each column and at most two 1's in each row. Noting that any column permutation preserves the consecutiveness of 1's in a row with at most one 1, we only focus on the (3, 2)-matrices which have exactly two 1's in each row in the rest of this section.

Let A be such a (3, 2)-matrix. Then, it defines uniquely a graph $G(A)$ with maximum degree 3 in which nodes correspond one-to-one to the columns in A and edges to the rows in A. Without loss of generality, we assume A have m row and n columns and the corresponding graph $G(A)$ has node set $V = \{1, 2, \cdots, n\}$. Furthermore, we have the following fact.

Lemma 3. *Let $C = \{i_1, i_2, \cdots, i_k\}$ be a subset of columns of A. Then, the submatrix $A(C)$ of A consisting of the columns in C has the C1P property if and only if the subgraph $G(A)|_C$ induced by node subset $C \subseteq V$ is an union of paths. Here we consider an isolated note as a trivial path.*

Proof. Assume $A(C)$ has the C1P property. Consider a node $i' \in C$. If it has three adjacent nodes i_j, i_k, i_l. Then, any permutation of C cannot keep the 1's consecutive on the rows corresponding to (i', i_j), (i', i_k) and (i', i_l) since there are only two 1's in each row. Thus, each node in $G(A)|_C$ has degree at most 2. Similarly, we can also show that $G(A)|_C$ does not contains any cycles. Therefore, the node induced subgraph is an union of paths and isolated nodes.

Conversely, if $G(A)|_C$ is an union of paths, then if we arrange all the columns in each path together in the same order as they appear in the path, the resulting matrix have 1's consecutive on each row. This finishes the proof. □

Using this lemma, we are able to present a 0.5-approximation algorithm for (3, 2)-matrices.

Theorem 5. *There is a linear-time algorithm that always outputs a C1P submatrix consisting of at least $n/2$ columns given a $(3, 2)$-matrix A with n columns.*

Proof. Let A be a (3, 2)-matrix. Recall that we assume that each row contains exactly two 1's in A. By Lemma 3, we only need to find a subset C containing at least $n/2$ nodes in $G(A)$ such that the induced subgraph $G(A)|_C$ is an union of paths and isolated nodes. The following Algorithm B is such an algorithm.

```
Algorithm B

Input:  A (3, 2)-matrix $A$;
Output: A subset $C$ of columns of $A$ such that $A(C)$ has the C1P property.

1    Construct the graph $G(A) = (V, E)$ as described above;
     ($G(A)$ has $A^t$ as its incidence matrix; each node has degree at most 3.)
2    Initially, set $V' = \phi$ and $V'' = V$;
3    Repeat the following action until both $G(A)|_{V'}$ and $G(A)|_{V''}$ contain
     no nodes of degree more than 1;
4        Pick a node of degree at least 2 in $G(A)|_{V'}$ and move it to $V''$, or
5        Pick a node of degree at least 2 in $G(A)|_{V''}$ and move it to $V'$;
6    Output $C = V'$ if $|V'| \geq n/2$ or $V''$ otherwise.
```

Each execution of line 3-5 increases the number of cut edges between V' and V'' by at least 1; and hence it will repeat at most $|E|$ times. Therefore, the algorithm takes linear time. Since $G(A)|_{V'}$ and $G(A)|_{V''}$ contains no nodes with degree more than 1, by Lemma 3, the output C is a desired subset of columns, i.e. $A(C)$ has the C1P property. This finishes the proof. □

4.2 Generalization to (Δ, 2)-Matrices

Given a graph G, we use $\Delta(G)$ to denote the largest degree of a node in G. Algorithm B indicates that the node set V of any graph G with $\Delta(G) = 3$ can be partitioned into V' and V'' such that $\Delta(G|_{V'}) \leq 1$ and $\Delta(G|_{V''}) \leq 1$. In fact, this is a special case of the following important partition theorem by setting $t_1 = t_2 = 1$.

Theorem 6. *([17]) Let $G = (V, E)$ be a graph. Let t_1, t_2, ..., t_k be non-negative integers such that $\sum_{i=1}^{k} (t_i + 1) - 1 = \Delta(G)$. Then V can be partitioned into k subsets that induce subgraphs G_1, G_2, ..., G_k with $\Delta(G_i) \leq t_i$, for $i = 1, 2, ..., k$.*

In addition, a desired partition can also be found in polynomial time [13]. By this theorem, the node set of a graph G can be decomposed into $\lceil(\Delta(G)+1)/2\rceil$ subsets that induce subgraphs with maximum degree at most 1. This implies the following result.

Proposition 3. *There is a polynomial time algorithm that always outputs a subset C of at least $n/\lceil(\Delta+1)/2\rceil$ columns such that $A(C)$ has the C1P property given an input (Δ, 2)-matrix A of n columns.*

5 Conclusion

The COS problem finds applications in physical mapping with hybridization date. In this paper, we answer an open problem posed in [12] by proving that the decision version of the COS problem remains NP-complete for $(2, 3)$ and $(3, 2)$-matrices. To prove these results, we also formulate two simple, but interesting NP-complete problems for cubic graphs. These two problems - Spanning Caterpillar Tree and Induced Disjoint-Path-Union Subgraph may find applications in studying the complexity issues of other algorithmic problems.

We also study the approximation issue of the COS problem. It is proved that the COS problem is 0.8-approximatable for $(2, 3)$-matrices and 0.5-approximatable for the matrices in which each column contains at most two 1's and for $(3, 2)$-matrices. But it is open whether the COS problem can be approximatable with constant factor for matrices in which there are at most two 1 in each row.

By studying the complexity and approximation issues of the COS problem that are relevant for physical mapping, we hope our results will give insights into the difficulty of the physical mapping problem which are of value for bioinformaticians.

References

1. J. Atkins and M. Middendorf. On physical mapping and the consecutive ones property for sparse matrices. *Discrete Applied Mathematics*, **71** (1996), 23-40.
2. F. Alizadeh, R. M. Karp, D. K. Weisser and G. Zweig. Physical mapping of chromosomes using unique probes. *Journal of Computational Biology*, **2** (1995), 159-184.
3. K. S. Booth and G. S. Lueker. Test for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Systems Sci.*, **13** (1976), 335-379.
4. J. S. Deogun and K. Gopalakrishnan. Consecutive retrieval property revisited. *Information Processing Letters*, **69** (1999), 15-20.
5. M. Flammini, G. Gambosi and S. Salomone. Boolean routing. *Lecture Notes in Comput. Sci.*, **725** (1993), 219-233.
6. S. Foote, D. Vollrath, A Hilton and D. C. Page. The human Y chromosome: overlapping DNA clones spanning the euchromatic region. *Science*, **258** (1992), 60-66.
7. D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Mathematics*, **15** (1965), 835-855.
8. M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* San Francisco: W. H. Freeman, 1979.
9. S. P. Ghosh. File organization: the consecutive retrieval property. *Commun. ACM*, **15** (1972), 802-808.
10. D. S. Greenberg and S. Istrail. Physical mapping by STS hybridization: algorithmic strategies and the challenge of software evaluation. *J. Comput. Biol.*, **2** (1995), 219-273.
11. M. Habib and R. 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** (2000), 59-84.
12. M. T. Hajiaghayi and Y. Ganjali. A note on the consecutive ones submatrix problem. *Information Processing Letters*, **83** (2002), 163-166.

13. M. M. Halldórsson and H. C. Lau. Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and 3-coloring. *J. Graph Algorithm Appl.*, **1** (3) (1997) 1-13.
14. W.-F. Lu and W.-L. Hsu. A test for the consecutive ones property on noisy data - application to physical mapping and sequence assembly. *Journal of Computational Biology* **10(5)** (2003), 709-735.
15. D. G. Kendall. Incidence matrices, interval graphs and seriation in archaeology. *Pacific J. Math.*, **28** (1969), 565-570.
16. J. M. Lewis. On the complexity of the maximum subgraph problem. in *Proc. 10th Ann. ACM Symp. on Theory of Computing*, (1978) 265-274.
17. L. Lovász. On decomposition of graphs. *Stud. Sci. Math. Hung.*, **1** (1966), 237-238.
18. J. Meidanis, O. Porto and G. P. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, **88** (1998), 325-354.
19. R. Mott, A. Grigoriev, and H. Lehrach. A algorithm to detect chimeric clones and randome noise in genomic mapping. *Genetics*, **22** (1994), 482-486.
20. P. A. Pevzner. *Computational molecular biology.* The MIT Press, 2000.
21. S. Weis and R. Reischuk. The complexity of physical mapping with strict chimerism. in *Proc. 6. Int. Symposium on Computing and Combinatorics (COCOON'2000)*, LNCS **1858**, 383-395.
22. M. Yannakakis. Node- and edge-deletion NP-complete problems. in *Proc. 10th Ann. ACM Symp. on Theory of Computing*, (1978) 253-264.