The Complexity of the Gapped Consecutive-Ones Property Problem for Matrices of Bounded Maximum Degree

*JÁN MAŇUCH^{1,2} and *MURRAY PATTERSON²

ABSTRACT

The Gapped Consecutive-Ones Property (C1P) Problem, or the (k, δ) -C1P Problem is: given a binary matrix M and integers k and δ , decide if the columns of M can be ordered such that each row contains at most k blocks of 1's, and no two neighboring blocks of 1's are separated by a gap of more than δ 0's. This problem was introduced by Chauve et al. (2009b). The classical polynomial-time solvable C1P Problem is equivalent to the $(1, 0)$ -C1P problem. It has been shown that, for every unbounded or bounded $k \geq 2$ and unbounded or bounded $\delta \geq 1$, except when $(k, \delta) = (2, 1)$, the (k, δ) -C1P Problem is NP-complete (Maňuch et al., 2011; Goldberg et al., 1995). In this article, we study the Gapped C1P Problem with a third parameter d , namely the bound on the maximum number of 1's in any row of M , or the bound on the maximum *degree* of M . This is motivated by the reconstruction of ancestral genomes (Ma et al., 2006; Chauve and Tannier, 2008), where, in binary matrices obtained from the experiments of Chauve and Tannier (2008), we have observed that the majority of the rows have low degree, while each high degree row contains many rows of low degree. The (d, k, δ) -C1P Problem has been shown to be polynomial-time solvable when all three parameters are fixed (Chauve et al., 2009b). Since fixing d also fixes k ($k \leq d$), the only case left to consider is the case when δ is unbounded, or the (d, k, ∞) -C1P Problem. Here we show that for every $d > k \geq 2$, the (d, k, ∞) -C1P Problem is NP-complete.

Key words: ancestral genome, reconstruction, computational complexity, consecutive-ones property, hypergraph covering.

1. INTRODUCTION

LET M BE A BINARY MATRIX WITH m ROWS AND n COLUMNS. A *block* in a row of M is a maximal sequence of consecutive entries containing 1. A *gap* is a sequence of consecutive 0's that separates two blocks, where the size of a gap is the length of this sequence of 0's. The *degree* of a row of M is the number of 1's in the row. Matrix M is said to have the *Consecutive-Ones Property* (C1P) if its columns can be permuted such that each row contains one block (there are no gaps in this case). We call a permutation π of the columns of M

¹Department of Mathematics, Simon Fraser University, Burnaby, Canada.

²Department of Computer Science, University of British Columbia, Vancouver, Canada.

*The two authors have contributed equally.

that witnesses this property a *consecutive-ones order* of M . Deciding if a binary matrix has the C1P can be done in linear time (Booth and Lueker, 1976).

Among its many applications, the C1P has been widely used in molecular biology, in relation to physical mapping (Alizadeh et al., 1995; Atkins and Middendorf, 1996) and the reconstruction of ancestral genomes (Ma et al., 2006; Chauve and Tannier, 2008). In the reconstruction of ancestral genomes in particular, we are given a set of extant species and a phylogenetic tree on this set, and the goal is to infer possible genomes (or gene orders) of an ancestor at some internal node higher up in this phylogenetic tree. This problem is modeled, for example in Chauve and Tannier (2008), by a binary matrix M as follows. Each column of M represents a gene or (homologous) genomic marker (from the collection of genomic markers common to the set of extant descendant species), while each row of the matrix represents a subset of markers that is believed to be contiguous (this subset appears contiguous in at least two extant species, for example), called a *synteny*, in this ancestor. Any consecutive-ones order of M corresponds then to a potential genome (or gene order) of this ancestor. A common problem in such applications, however, is that matrices obtained from experiments do not have the C1P, often due to small errors in the data (Goldberg et al., 1995; Weis and Reischuk, 2000; Lu and Hsu, 2003; Chauve and Tannier, 2008). It is the lack of robustness of the C1P to small perturbations that led researchers to consider the following approaches for handling matrices that do not have the C1P.

A first general approach for handling a matrix M that does not have the C1P consists of transforming M into a matrix that has the C1P, while minimizing the modifications to M ; such modifications can involve either removing rows, or columns, or both, or flipping some entries from 0 to 1 or 1 to 0. In all cases, the corresponding optimization problems have been proven to be NP-complete (Hajiaghayy and Ganjali, 2002; Dom et al., 2007; Dom, 2008).

A second general approach is to allow the columns of M to appear more than once in a consecutive-ones order of M . Such an approach was first introduced by Wittler and Stoye (2010), where the authors defined the following problem: given binary matrix M on columns $C = \{1, \dots, |C|\}$ and function $\mathbf{m} : C \rightarrow \mathbb{N}$, is there a *sequence* s over alphabet C such that (i) s contains each column $c \in C$ at most $\mathbf{m}(c)$ times, and (ii) for each row r of M , the set of columns that have entry 1 in r form at least one subsequence of s . Wittler and Stoye (2010) show that this problem is polynomial-time solvable if each row of M has at most two 1's, while if each row of M has at most five 1's, the problem, and two restricted variants thereof, are NP-complete. In Wittler et al. (2011), this NP-completeness result was improved to each row of M having at most three 1's and each column appearing at most twice, using a reduction from a new version of hypergraph covering problem that was inspired by the results of this work.

Here we propose several versions of a third approach to this problem, motivated by several scenarios within the reconstruction of ancestral genomes setting; however, similar motivations could be found in other applications; see Goldberg et al. (1995) for one example. The third general approach consists of relaxing the consecutivity condition of the 1's in each row by allowing gaps with some restriction on the nature of these gaps. The question is then to decide if there is an ordering of the columns of M that satisfies these relaxed C1P conditions. Goldberg et al. (1995) introduced the notion of the *k-Consecutive-Ones Property* (k -C1P). A binary matrix M has the k -C1P when its set of columns can be permuted such that each row contains at most k blocks. This version of the problem does not put any restriction on the size of the gaps between blocks. This version of the problem pertains to the reconstruction of ancestral genomes in that an observed synteny can consist of two or more syntenies if they were made contiguous by independent rearrangements in two (or more) species, which is referred to as a *chimerism*. This might happen especially in genomes such as yeasts where we generally see many translocations (Jinks-Robertson and Petes, 1986; Richardson and Jasin, 2000). Goldberg et al. (1995) show that deciding if a binary matrix M has the k -C1P is NP-complete, even if $k = 2$. Also, finding an ordering of the columns that minimizes the number of gaps in M is NP-complete, even if each row of M has at most two 1's (Haddadi, 2002).

Chauve et al. (2009b) then defined the Gapped C1P Problem, or the (k, δ) -C1P Problem: given binary matrix M and two integers k and δ , to decide if the columns of M can be ordered such that each row contains at most k blocks, and no two neighboring blocks of 1's are separated by a gap of size more than δ . This version of the problem is motivated by a different scenario. Even though a group of genomic markers is expected to appear together in the ancestral genome, over evolution, insertion among this group of several markers from outside this group could happen, resulting in a small gap. Another reason for considering small gaps is to handle errors in defining (homologous) genomic markers (paralogies inferred instead of orthologies in constructing this set of markers), errors from convergent evolution, etc., when

these errors tend to be small. Finally, we note that, in the model of max-gap clusters (Pasek et al., 2005), another form of generalization of common intervals (i.e., the C1P), gaps, are of bounded size. So if ancestral syntenies are detected or inferred under this model, then the resulting matrix should have the (k, δ) -C1P for bounded δ . In Mañuch et al. (2011), it was shown that for every $k \geq 2$, $\delta \geq 1$, $(k, \delta) \neq (2, 1)$, the (k, δ) -C1P Problem is NP-complete, leaving open only the complexity of the $(2, 1)$ -C1P case. The problem remains NP-complete even if one of the two parameters is unbounded: (i) for every $k \geq 2$, the (k, ∞) -C1P Problem is just the problem of deciding if a matrix M has the k -C1P, and is thus NP-complete by Goldberg et al. (1995), and (ii) for every $\delta \geq 1$, the (∞, δ) -C1P Problem is NP-complete (Mañuch et al., 2011).

The above NP-completeness results on the (k, δ) -C1P Problem involve constructions with many rows of high degree. After examining some data from experiments, we found that this is not always realistic. We considered here the ancestral syntenies dataset for the boreoeutherian ancestor of Chauve and Tannier (2008) at a resolution of 200 kb, with 1651 markers (i.e., columns) and 2515 syntenies (rows).¹ In this dataset, we observed that 90% of the syntenies have low degree (less than or equal to 16, which is less than 1% of the number of columns of this matrix). Each of the remaining 10% of the syntenies (with degrees 17–99) contains between 16–144 syntenies with low degree. This is in agreement with the theoretical observations of Bouvel et al. (2009), where it was shown that the rows/syntenies with high degree with high probability contain some rows/syntenies with low degree. Hence, if the syntenies with high degree (10%) are discarded, the majority of the information is preserved, and so it makes sense to consider versions of the problem where the degree is bounded, especially, if this could help to turn an NP-complete problem to polynomial.

Formally, we have the (d, k, δ) -C1P Problem: given matrix M where the bound on the maximum degree of any row of M is d , decide if it has the (k, δ) -C1P. We call a permutation π of the columns of M that witnesses this property a (d, k, δ) -consecutive order; that the matrix M' resulting from this permutation is (d, k, δ) -consecutive, or that it is consecutive w.r.t. π ; and that M is (d, k, δ) -C1P, or has the (d, k, δ) -C1P. If all three parameters are fixed, the problem is related to the classical Graph Bandwidth Problem, and can be solved in polynomial time using a variant of an algorithm of Saxe (Saxe, 1980; Chauve et al., 2009b). However, this is practical only for very small values of the three parameters.

In this article, we study the complexity of the (d, k, δ) -C1P Problem when one or more of these parameters are unbounded. The cases with d unbounded were considered in Mañuch et al. (2011). Since fixing d also fixes k ($k \leq d$), the only case left to consider is the case when δ is unbounded, or the (d, k, ∞) -C1P Problem. This case deals with the problem of chimerisms and assumes that we do not lose too much information by considering only syntenies with low degree as argued above. Here we show that in every non-trivial case, this problem is NP-complete, i.e., for every $d > k \geq 2$, the (d, k, ∞) -C1P Problem is NP-complete. Note that if $d = 2$, the problem becomes the C1P Problem, and if $d \leq k$, then any order of the columns of M is a valid solution, since no row can have more than d blocks of 1's.

This article is structured as follows. First, in Section 2, we give the definition of a type of hypergraph covering problem. In Section 3, we show that a special case of this hypergraph covering problem is NP-complete, and then in Section 4, we generalize this construction to show that the general case of this hypergraph covering problem is NP-complete. Finally, in Section 5, we show a direct correspondence of the general case of this hypergraph covering problem to the (d, k, ∞) -C1P Problem for every $d > k \geq 2$ to give the result of this work. In Section 6, we conclude the article with some remarks and discuss future work.

2. A HYPERGRAPH COVERING PROBLEM

We first define the following hypergraph covering problem. In the sections that follow, we will show that this problem is NP-complete, and that it corresponds exactly to the (d, k, ∞) -C1P Problem for the result of this article. Note that a hypergraph $H = (V, E)$ is d -uniform when all its hyperedges are d -edges, that is, hyperedges that contain exactly d vertices.

Definition 1 (p -Covering of a d -Uniform Hypergraph). Given a d -uniform hypergraph $H = (V, E)$ and an integer p , let $K_{|V|}$ be a complete graph on V and let $\mathcal{P}_p$ be the set of all subsets of $E(K_{|V|})$ with exactly p edges. A p -covering of H is a graph $G = (V, E')$ such that there exists a map $\mathbf{c} : E \rightarrow \mathcal{P}_p$ such that

¹This dataset can be found at www.cecm.sfu.ca/~cchauve/SUPP/ANCESTOR08/BOREO_200_u/index.html.

- for every $h \in E$, and for every $e \in \mathbf{c}(h)$, $e \subseteq h$; and
- $E' = \bigcup_{h \in E} \mathbf{c}(h)$.

Here, we say that set $\mathbf{c}(h)$ p -covers the hyperedge h and that G p -covers H .

Informally, a p -covering of a d -uniform hypergraph is a graph constructed by picking p edges from each hyperedge.

Problem 1 (d -Uniform Hypergraph p -Covering by Paths Problem [d -UH- p -CP Problem]). *Given a d -uniform hypergraph $H = (V, E)$ and an integer $p < d$, is there a p -covering of H which consists only of disjoint paths?*

Variations of this problem were defined in Gupta et al. (2007, 2008, 2009). The first variation allowed the hypergraph to have only 2-, 3-, and 4-edges, where 2- and 3-edges were covered by picking one edge, while 4-edges were covered by two parallel edges, and required that the covering contains only disjoint edges and vertices. This variation was shown to be polynomial-time solvable which provided an algorithm for a special version of haplotyping problem via galled-tree networks (Gupta et al., 2007). The second variation allowed only 3-uniform hypergraphs, and required all connected components of the covering to be paths of length at most 3. This variation was shown to be NP-complete (Gupta et al., 2009). A slightly more complex version of this was then used to show that, in general, the haplotyping problem via galled-tree networks is NP-complete (Gupta et al., 2008).

In the next section, we show that the special case, namely the 3-UH-1-CP Problem is NP-complete, which is then generalized in Section 4 to show NP-completeness of the d -UH- p -CP Problem for every $d - 2 \geq p \geq 1$.

3. THE 3-UNIFORM HYPERGRAPH 1-COVERING BY PATHS PROBLEM

We now show that the 3-Uniform Hypergraph 1-Covering by Paths (3-UH-1-CP) Problem is NP-complete.

Theorem 1. *The 3-UH-1-CP Problem is NP-complete.*

Proof. Clearly, the problem is in NP. We will show it is also NP-hard by reduction from 3SAT(3), a restricted version of 3SAT, proved NP-complete by Papadimitriou (1994), in which every variable has exactly two positive and one negative occurrence in the clauses.² We will call a p -covering of a hypergraph *valid* if it consists only of disjoint paths. Note that a valid p -covering does not contain vertices of degree 3 or more and does not contain cycles. Given 3SAT(3) formula ϕ with variables $X = \{x_1, \dots, x_n\}$ and clauses $C = \{c_1, \dots, c_m\}$, we now construct a 3-uniform hypergraph H_ϕ on at most $12n + 15m$ hyperedges which contains, among other vertices, a vertex for each literal of ϕ (there are $3n$ such vertices) that has a valid 1-covering if and only if ϕ is satisfiable.

First we give an important building block that is used throughout this construction: the complete 3-uniform hypergraph D on 4 vertices. In any valid 1-covering of D , there is no isolated vertex. Indeed, assume for contradiction that v is the isolated vertex in a valid covering G of D . Let u_1, u_2, u_3 be the remaining three vertices. Then there is a pair u_i, u_j such that $\{u_i, u_j\}$ is not an edge in G . However, no edge is 1-covering hyperedge $\{v, u_i, u_j\}$, a contradiction. We will use several copies of D in the construction to introduce a dependency on 1-coverings of touching hyperedges and depict them as diamonds in the figures. For instance, consider the hypergraph in Figure 1a. Since in any valid 1-covering G of this hypergraph, v is a member of an edge in D , at most one of the hyperedges h_1 and h_2 can “pick” an edge involving v , otherwise vertex v would have degree 3 or more.

Now to the main construction. Consider the instance ϕ of 3SAT(3) with variables $X = \{x_1, \dots, x_n\}$ and clauses $C = \{c_1, \dots, c_m\}$. In the construction, any valid covering selects a set of literals (more precisely,

²We remark that the exact formulation of 3SAT(3) in Papadimitriou (1994) allows also variables with one positive and two negated occurrences, but these can easily be converted to the other type of variables by replacing them with their negations in all clauses. Clearly, this does not affect the complexity of the problem.

FIG. 1. (a) A simple dependency on 1-coverings of two touching hyperedges enforced by a copy of D (depicted as a diamond). (b) The 2-clause and (c) 3-clause gadgets for clause c_i .

the vertices corresponding to these literals), i.e., positive and negative occurrences of variables. If this selection satisfies the following two properties:

- (1) every clause selects at least one literal, and
- (2) for every $x \in X$, at most one of x and $\neg x$ is selected,

then this selection can be used to build a satisfying truth assignment for ϕ as follows: for every $i \in \{1, \dots, n\}$ if x_i ($\neg x_i$) is in the selection, set the value of x_i to *true* (*false*). If neither x_i and or $\neg x_i$ is in the selection, pick the value at random. We design a hypergraph H_ϕ composed of clause gadgets which will guarantee the first condition and variable gadgets which will ensure the second condition.

Figure 1b,c depicts the 2-clause and 3-clause gadgets, respectively. Given a valid 1-covering G of the clause gadget for clause c_i with literals c_i^1 , c_i^2 (and c_i^3 for a 3-clause), we say that a literal vertex c_i^j is *selected* in G , if c_i^j is contained in two edges of the covering G . Note that in both clause gadgets at least one of the literal vertices is selected in any valid covering. This is obvious for the 2-clause gadget. For the 3-clause gadget, if none of the literal vertices is selected in a valid 1-covering of this gadget, then in the three hyperedges in Figure 1c, no picked edge involves c_i^1 , c_i^2 or c_i^3 . But this creates a cycle, a contradiction. Now, each literal vertex c_i^j will also appear in exactly one variable gadget described in the next paragraph. If a literal vertex c_i^j is selected in a valid covering, then it cannot be contained in any edge that covers the hyperedges of the variable gadget; otherwise, c_i^j has degree 3 or more in this covering. The variable gadget for each $x \in X$ will use this property to ensure that literal vertices x and $\neg x$ are not selected at the same time.

Figure 2a depicts the variable gadget for variable $x \in X$ with the two positive occurrences c_i^p and c_j^q , and one negated occurrence c_k^r of this variable x in the clauses. Note that if both a positive and the negated literal vertices of x are selected by a clause gadget in a valid 1-covering of H_ϕ , then it forces a cycle in the variable gadget of x , a contradiction. It follows that if H_ϕ has a valid 1-covering then ϕ is satisfiable.

Conversely, if ϕ has a satisfying assignment τ , let us pick one literal for each clause which makes it satisfied in τ and build the 1-covering of H_ϕ as follows. In each clause gadget: (i) in each hyperedge of this clause gadget that contains a literal vertex, pick an edge containing the literal vertex if this literal was selected for this corresponding clause; and (ii) for each diamond, choose any of the 3 valid 1-coverings of this diamond that consist of 2 parallel edges. In the variable gadgets, pick the edges as depicted in Figure 2b if the variable has value *false* in τ , and otherwise, pick the edges as depicted in Figure 2c. By selecting edges in this fashion, every hyperedge of H_ϕ is 1-covered by an edge, and each literal vertex is adjacent to

FIG. 2. (a) The variable gadget for variable with positive occurrences c_i^p and c_j^q and negated occurrence c_k^r in the clauses. The dashed edge is always picked in any valid 1-covering. (b) Gray edges are picked when this variable is set to *false* in a satisfying assignment of ϕ . (c) Gray edges are picked when the variable is set to *true*.

at most two edges in the 1-covering, one of them lying in the diamond. Hence, there is no vertex of degree 3 and no cycles in this 1-covering, i.e., this 1-covering is valid.

Since the number of hyperedges used in the construction is at most $15m + 12n$, i.e., linear in the size of ϕ , this construction can be built in polynomial-time, and hence, the 3-UH-1-CP Problem is NP-hard. ■

In the following section, we generalize this construction to show that, for every $d - 2 \geq p \geq 1$, the d -UH- p -CP Problem is NP-complete.

4. THE d -UNIFORM HYPERGRAPH p -COVERING BY PATHS PROBLEM

We now show how the construction of Section 3 can be generalized to show that for every $d - 2 \geq p \geq 1$, the d -Uniform Hypergraph p -Covering by Paths (d -UH- p -CP) Problem is NP-complete. The main building block in this new construction is the following d -uniform hypergraph that generalizes the hypergraph D (the diamond) from the previous construction of Section 3.

Lemma 1. *For any $d - 2 \geq p \geq 1$, there exists a d -uniform hypergraph $D_{d,p} = (V, E)$ with a distinguished vertex $v \in V$ that has the following properties:*

1. $|V| = 2d - p - 1$ and $|E| = \binom{2(d-p)-1}{d-p}$;
2. in any valid p -covering of $D_{d,p}$, v is not isolated; and
3. hypergraph $D_{d,p}$ has a valid p -covering in which v has degree 1.

Proof. Let $D_{d,p} = (V, E)$ be the d -uniform hypergraph on the vertex set $V = S \cup P \cup \{v\}$ where $|S| = 2(d - p) - 1$, $|P| = p - 1$, and v is the single distinguished vertex. For every subset $S' \subseteq S$ of size $d - p$, we add a hyperedge on the d vertices $S' \cup P \cup \{v\}$ to E , i.e., $|E| = \binom{2(d-p)-1}{d-p}$. Hypergraph $D_{d,p}$ is depicted in Figure 3. We now show that this hypergraph satisfies conditions 2 and 3 of the lemma. Here, again, we call a graph p -covering of a hypergraph *valid* if it consists only of disjoint paths.

Assume, for contradiction, that v is isolated in a valid p -covering G of $D_{d,p}$. Since G is some collection of paths on the vertex set $S \cup P$, *virtual* edges can always be added to G to extend this collection to a single path G' on this set. In what follows, we will find a hyperedge in $D_{d,p}$ and show that it contains less than p edges in G' , and hence, less than p edges in G , and thus, is not covered by G .

Path G' defines a total order on its vertex set $S \cup P$ (there are two such total orders, but we can choose either one, without loss of generality). Let $t : S \cup P \rightarrow \mathbb{N}$ be this total order. If we follow the vertices of path G' according to t , it starts at some vertex in one of S or P , alternates between the two sets, and then terminates in one of these sets. Hence, the subgraph G'_S (resp. G'_P) of G' induced on vertex set S (resp. P) is some collection of paths on S (resp. P), say $S_1, \dots, S_r$ (resp. $P_1, \dots, P_\ell$), where for any $i < j$, vertex $u \in S_i$ (resp. P_i) and $u' \in S_j$ (resp. P_j), $t(u) < t(u')$ (Fig. 4).

Let us order the elements of S according to total order t and let S' be the odd numbered elements of S according to this order. Since $|S| = 2(d - p) - 1$, $|S'| = d - p$. Now, consider the d -edge $h = S' \cup P \cup \{v\}$ of

FIG. 3. Hypergraph $D_{d,p}$: only one of the $\binom{|S|}{d-p}$ hyperedges is shown.

FIG. 4. The path G' through vertex set $S \cup P$ that alternates between subpaths completely in S and completely in P . Some of the shown edges may be virtual.

$D_{d,p}$. Hyperedge h is indeed an edge in $D_{d,p}$ since it contains $P \cup \{v\}$ and a subset, namely S' , of size $d - p$ of S . Hyperedge h for the example of Figure 4 is depicted in Figure 5. We will show that this hyperedge contains less than p edges from G' .

Let us count the number of edges of G' that are contained in h . Each path P_i , $i = 1, \dots, \ell$ is completely contained in h , and thus contributes to h the $|P_i| - 1$ edges that connect the vertices of this path. On the other hand, since S' is the set of odd numbered elements of S according to total order t , none of the edges in S_j , $j = 1, \dots, r$ is contained in h . Finally, we need to consider edges of the path G' crossing between the sets S and P . We will show that for each $i = 1, \dots, \ell$, there is at most one crossing edge starting at a vertex of P_i and ending in S that is contained in h . There are at most two edges starting at a vertex of P_i and ending in some vertex of S . If the number of these edges is less than two, the claim holds. Assume there are two such edges. They must start at the endpoints of P_i and end in the consecutive elements of S (according to t). Hence, at most one of them is ending in the odd numbered element of S , i.e., contained in h . It follows that the number of crossing edges contained in h is at most ℓ . Hence, h contains at most $\ell + \sum_{i=1}^{\ell} (|P_i| - 1) = |P| = p - 1$ edges of G' , and hence, at most $p - 1$ edges of G , thus it is not p -covered by G , a contradiction. We can conclude that in any valid p -covering of $D_{d,p}$, vertex v has degree at least one.

Finally, we show that $D_{d,p}$ has a valid p -covering in which vertex v has degree 1. Consider the path G that starts at v and then visits all vertices in P and then all vertices in S (Fig. 6). Consider any hyperedge $h = S' \cup P \cup \{v\}$ where S' is some subset of S of size $d - p$. The hyperedge h contains $P \cup \{v\}$, and thus the subpath of G induced by these vertices. This subpath has $p - 1$ edges. Consider the subgraph of G induced

FIG. 5. Hyperedge h of $D_{d,p}$ which contains less than p edges from G' depicted in Figure 4.

FIG. 6. A valid p -covering of $D_{d,p}$ in which vertex v has degree 1.

by S' . If this subgraph contains at least one edge, we pick this edge for h , and hence, h is p -covered by G . Otherwise, S' must consist only of odd numbered elements of the subpath of G induced by S , and thus it contains the first vertex of this subpath. Hence, h contains the edge of G connecting sets S and P , and we pick this edge for h , i.e., it is p -covered by G . ■

In the following theorem, we will use many copies of $D_{d,p}$ to simulate the behavior of a 3-edge in the 1-covering problem with a d -edge in the p -covering problem.

Theorem 2. For every $d - 2 \geq p \geq 1$, the d -UH- p -CP Problem is NP-complete.

Proof. Clearly, this problem is in NP. We will show that it is also NP-hard by reduction from the 3-UH-1-CP Problem that was shown to be NP-complete in Section 3.

Given a 3-uniform hypergraph $H = (V, E)$, we will construct a d -uniform hypergraph $\bar{H}$ that has a valid p -covering if and only if H has a valid 1-covering. For each 3-edge $h = \{a, b, c\} \in E$ we add the corresponding d -edge $\bar{h} = \{a, b, c, h_1, \dots, h_{d-3}\}$ to $\bar{H}$. To simulate in $\bar{H}$, the behavior of h , we then add $2(d - p - 2)$ copies of $D_{d,p}$ to $\bar{H}$, where the distinguished vertex v of each copy is identified with one of the vertices $h_p, \dots, h_{d-3}$ such that each of them is used exactly twice. Figure 7 illustrates all vertices and hyperedges added to $\bar{H}$ for this 3-edge h in H . We note that all vertices other than a, b, c added to $\bar{H}$ for h are disjoint from all other vertices.

Now, assume that there is a valid p -covering $\bar{G}$ of $\bar{H}$. We will construct a 1-covering G of H as follows. For each $h \in E$, consider the subgraph $\bar{G}_{\bar{h}}$ of $\bar{G}$ induced by the vertices in $\bar{h}$. It must have at least p edges. By Lemma 1, vertices $h_p, \dots, h_{d-3}$ are incident to some edges of $\bar{G}$ in two different diamonds, and since

FIG. 7. Vertices and hyperedges added to $\bar{H}$ to simulate the 3-edge $h = \{a, b, c\}$. The grayed diamonds depict copies of $D_{d,p}$.

there is no vertex of degree 3 in $\bar{G}$, they are isolated vertices in $\bar{G}_h$. Hence, we have p edges in the $p+2$ element set $\{a, b, c, h_1, \dots, h_{p-1}\}$ which cannot create a cycle. It hence follows that these vertices must form at most two components. Therefore, at least one pair of the vertices a, b, c must lie in the same component. If there is only one such pair, we add it to G as an edge. If all three vertices a, b, c are connected, we add to G a pair which remains connected after removing the third vertex. As a consequence of this choice, each edge $\{u, v\}$ in G covering a hyperedge h in H corresponds to a path in $\bar{G}$ connecting u and v . In addition, all internal vertices of these paths are not in V , and since hyperedges in $\bar{H}$ share only vertices in V , they are pairwise internally vertex disjoint.

The graph G constructed above is obviously a 1-covering of H . Let us check that it is also valid. First, if there is a vertex $u \in V$ with degree 3 or more, then there are three internally disjoint paths starting at u in $\bar{G}$, i.e., u would have degree at least 3 in $\bar{G}$, a contradiction. Second, if there is a cycle $u_1, u_2, \dots, u_k, u_1$ in G , then for each edge $\{u_i, u_{i+1}\}$ in G , we have a path connecting u_i and u_{i+1} in $\bar{G}$. Since these paths are internally vertex disjoint, they create a cycle in $\bar{G}$, a contradiction.

Conversely, assume there is a 1-covering G of H . We construct a p -covering $\bar{G}$ of $\bar{H}$ as follows. Cover each copy of $D_{d,p}$ such that the distinguished vertex has degree 1 (this is possible by Lemma 1). For each hyperedge $h = \{a, b, c\} \in E$, without loss of generality, let $\{a, b\}$ be the edge that covers h in G . Then cover hyperedge $\bar{h}$ by a path starting at a , visiting all vertices $h_1, \dots, h_{p-1}$ and ending at b , while the vertex c is an isolated vertex. This is a p -covering $\bar{H}$ and it is easy to verify that it is also valid.

Finally, let us check that the construction is polynomial. The number of vertices of $\bar{H}$ is $|V| + |E|[d - 3 + 2(d - p - 2)(2d - p - 2)]$ and the number of edges is $|E| \left[1 + 2(d - p - 2) \binom{2(d-p)-1}{d-p} \right]$.

Since d and p are assumed to be constants, the reduction is polynomial. ■

5. THE (d, k, ∞) -C1P PROBLEM

We now show that for every $d > k \geq 2$, the (d, k, ∞) -C1P is NP-complete, by showing the correspondence of this problem to the d -UH- $(d - k)$ -CP Problem. A d -uniform hypergraph $H = (V, E)$ can be represented as a binary matrix B_H with $|V|$ columns and $|E|$ rows, where for each hyperedge $h \in E$, we add a row with 1's in the columns corresponding to vertices in h and 0's everywhere else. Obviously, the degree of every row of B_H is d and there is a one-to-one correspondence between d -uniform hypergraphs and such matrices.

Lemma 2. *A d -uniform hypergraph $H = (V, E)$ can be $(d - k)$ -covered by disjoint paths if and only if matrix B_H has the (d, k, ∞) -C1P.*

Proof. Assume first that H has a valid covering G . Since G consists of disjoint paths, there is a Hamiltonian path P on V containing all edges of G . This path defines an order on the vertices in V . Consider the ordering of the columns of matrix B_H based on this order (V is the set of columns B_H). We will show that this ordering is (d, k, ∞) -consecutive. Since each row of B_H contains exactly d 1's, it is enough to show that $d - k$ pairs of these d columns are adjacent in this ordering. The d columns containing 1's in each row form a hyperedge in H . Since G is a valid $(d - k)$ -covering, there are edges between $d - k$ pairs of these d columns in G . Since P contains all edges of G , it contains also these $d - k$ edges and hence, each of the corresponding $d - k$ pairs of columns are adjacent in the ordering. It follows that the ordering of B_H is (d, k, ∞) -consecutive.

Conversely, assume that matrix B_H is (d, k, ∞) -consecutive. Let $\pi = v_{i_1}, \dots, v_{i_n}$ be the order of the columns in a (d, k, ∞) -consecutive ordering of B_H . Now, for any hyperedge $h = \{v_{j_1}, v_{j_2}, \dots, v_{j_d}\}$ of H , there is a row in B_H with 1's in these d columns, hence, $d - k$ pairs of the columns in h must be adjacent in the ordering π . Consider the following covering G of H : for every hyperedge pick the edge between each pair of adjacent columns/vertices. Note that every edge in G is $\{v_{i_j}, v_{i_{j+1}}\}$ for some j . Hence, G has no vertex of degree 3 or higher, nor any cycle, thus G is a collection of disjoint paths, i.e., a valid $(d - k)$ covering of H . ■

By Theorem 2 and Lemma 2, it follows that for every $d > k \geq 2$, the (d, k, ∞) -C1P Problem is NP-complete.

Theorem 3. *For every $d > k \geq 2$, the (d, k, ∞) -C1P Problem is NP-complete.*

6. CONCLUSION

In this work, we have studied the weakest formulation of the C1P Problem with gaps: indeed, in the $(d, d-1, \infty)$ -C1P case of the problem, it is required that only two of the d 1's in each row are adjacent, while the other 1's can end up arbitrarily far away from this pair. It is thus surprising that this problem is still NP-complete for any $d \geq 3$ as implied by the general result of this paper. This article closes the case of the complexity of the (d, k, δ) -C1P Problem, with the exception of the $(\infty, 2, 1)$ -C1P case, or just the $(2, 1)$ -C1P case (Mañuch et al., 2011), which remains open.

There are several directions we would like to follow in the future work: (i) Is it possible to find a nice characterization of non- (d, k, δ) -C1P matrices in terms of forbidden structures, such as Tucker submatrices (Tucker, 1972), especially, for small values of d ? It has recently been shown that such a characterization could be used in the design of algorithms related to the C1P (Dom, 2008; Chauve et al., 2009a; Blin et al., 2010). (ii) When all three parameters are fixed, the (d, k, ∞) -C1P is related to the classical Graph Bandwidth Problem, and thus can be solved in polynomial time (Chauve et al., 2009b) using a variant of a relatively brute-force algorithm of Saxe (1980) for deciding if a graph has bandwidth k . Caprara et al. (2002) provide a linear time algorithm for the special case of deciding if a graph has bandwidth 2. It would be useful to investigate if the algorithm of Caprara et al. (2002) can be extended for deciding bandwidth for small values $k \geq 2$, in an attempt to improve the algorithm of Chauve et al. (2009b) for deciding the (d, k, ∞) -C1P. (iii) The problem of covering hypergraphs with a collection of paths played a key role in the results of this paper article. Perhaps considering other conditions on the covering could give rise to other new interesting problems. (iv) Assuming that k is close to d , for each row there are many orders of columns which make this row (d, k, ∞) -consecutive. Hence, for a small number of rows, random instances of matrices have the (d, k, ∞) -C1P almost always. Conversely, for a large number of rows, random instances of matrices that have the (d, k, ∞) -C1P would have very few column orderings that witness this property. We would like to investigate the ratios between the number of rows and columns for which this is the case, with the goal of developing heuristics for both of these types of instances.

ACKNOWLEDGMENTS

We would like to thank Cedric Chauve for the helpful remarks on this article. Research was supported by a NSERC Discovery Grant (to J.M.) and NSERC PGS-D3 (to M.P.).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

- Alizadeh, F., Karp, R., Weisser, D., et al. 1995. Physical mapping of chromosomes using unique probes. *J. Comput. Biol.* 2, 159–184.
- Atkins, J., and Middendorf, M. 1996. On physical mapping and the consecutive ones property for sparse matrices. *Discr. Appl. Math.* 71, 23–40.
- Blin, G., Rizzi, R., and Vialette, S. 2010. A faster algorithm for finding minimum tucker submatrices. *Lect. Notes Comput. Sci.* 6158, 69–77.
- Booth, K.S., and Lueker, G.S. 1976. Testing for the consecutive ones property of, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.* 13, 335–379.
- Bouvel, M., Chauve, C., Mishna, M., et al. 2009. Average-case analysis of perfect sorting by reversals. *Lect. Notes Comput. Sci.* 5577, 314–325.
- Caprara, A., Malucelli, F., and Petrolani, D. 2002. On bandwidth-2 graphs. *Discr. Appl. Math.* 34, 477–495.
- Chauve, C., and Tannier, E. 2008. A methodological framework for the reconstruction of contiguous regions of ancestral genomes and its application to mammalian genomes. *PLoS Comput. Biol.* 4, e1000234.
- Chauve, C., Haus, U.-W., Stephen, T., et al. 2009a. Minimal conflicting sets for the consecutive-ones property in ancestral genome reconstruction. *Lect. Notes Bioinform.* 5817, 48–58.

- Chauve, C., Mañuch, J., and Patterson, M. 2009b. On the gapped consecutive-ones property. *Proc. EUROCOMB*, 121–125.
- Dom, M. 2008. Recognition, generation, and application of binary matrices with the consecutive-ones property [Ph.D. dissertation]. Institut für Informatik, Friedrich-Schiller-Universität, Jena.
- Dom, M., Guo, J., and Niedermeier, R. 2007. Approximability and parameterized complexity of the consecutive ones submatrix problems. *Lect. Notes Comput. Sci.* 4484, 680–691.
- Goldberg, P., Golumbic, M., Kaplan, H., et al. 1995. Four strikes against physical mapping of DNA. *J. Comput. Biol.* 2, 139–152.
- Gupta, A., Mañuch, J., Stacho, L., et al. 2007. Algorithm for haplotype inferring via galled-tree networks with simple galls. *Lect. Notes Bioinform.* 4463, 121–132.
- Gupta, A., Mañuch, J., Stacho, L., et al. 2008. Haplotype inferring via galled-tree networks is NP-complete. *Lect. Notes Comput. Sci.* 5092, 287–298.
- Gupta, A., Mañuch, J., Stacho, L., et al. 2009. Haplotype inferring via galled-tree networks using a hypergraph covering problem for special genotype matrices. *Discr. Appl. Math.* 157, 2310–2324.
- Haddadi, S. 2002. A note on the NP-hardness of the consecutive block minimization problem. *Int. Trans Oper. Res.* 9, 775–777.
- Hajiaghay, M.T., and Ganjali, Y. 2002. A note on the consecutive ones submatrix problem. *Inform. Process. Lett.* 83, 163–166.
- Jinks-Robertson, S., and Petes, T. 1986. Chromosomal translocations generated by high-frequency meiotic recombination between repeated yeast genes. *Genetics* 114, 731–752.
- Lu, W., and Hsu, W. 2003. A test for the consecutive ones property on noisy data—application to physical mapping and sequence assembly. *J. Comput. Biol.* 10, 709–735.
- Ma, J., Haussler, D., and Miller, W. 2006. Reconstructing contiguous regions of an ancestral genome. *Genome Res.* 16, 1557–1565.
- Mañuch, J., Patterson, M., and Chauve, C. 2011. Hardness results on the gapped consecutive-ones property problem. *Discr. Appl. Math.* (in press).
- Papadimitriou, C. 1994. *Computational Complexity*. Addison Wesley, New York.
- Pasek, S., Risler, J., Bergeron, A., et al. 2005. Identification of genomic features using microsynteny of domains: domain teams. *Genome Res.* 15, 867–874.
- Richardson, C., and Jasin, M. 2000. Frequent chromosomal translocations induced by DNA double-strand breaks. *Nature* 405, 697–700.
- Saxe, J.B. 1980. Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM J. Alg. Discr. Methods* 1, 363–369.
- Tucker, A.C. 1972. A structure theorem for the consecutive 1's property. *J. Combin. Theory B* 12, 153–162.
- Weis, S., and Reischuk, R. 2000. The complexity of physical mapping with strict chimerism. *Lect. Notes Comput. Sci.* 1858, 383–395.
- Wittler, R., and Stoye, J. 2010. Consistency of sequence-based gene clusters. *Lect. Notes Bioinform.* 6398, 252–263.
- Wittler, R., Mañuch, J., Patterson, M., et al. 2011. Consistency of sequence-based gene clusters. *J. Comput. Biol.* (in press).

Address correspondence to:
 Dr. Murray Patterson
 Department of Computer Science
 UBC
 Vancouver, BC V6T1Z4, Canada

E-mail: murray.patterson@gmail.com