On the Gapped Consecutive Ones Property

Cedric Chauve[1], Ján Maňuch[1,2], and Murray Patterson[2]

[1] Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada
[2] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

Abstract. Motivated by problems of comparative genomics and paleogenomics, we introduce the Gapped Consecutive-Ones Property Problem (k,δ)-C1P: given a binary matrix M and two integers k and δ, can the columns of M be permuted such that each row contains at most k sequences of 1's and no two consecutive sequences of 1's are separated by a gap of more than δ 0's. The classical C1P problem, which is known to be polynomial, is equivalent to the (1,0)-C1P Problem. We show that the (2,δ)-C1P Problem is NP-complete for $\delta \geq 2$. We conjecture that the (k, δ)-C1P Problem is NP-complete for $k \geq 2, \delta \geq 1, (k, \delta) \neq (2, 1)$. We also show that the (k,δ)-C1P problem can be reduced to a graph bandwidth problem parameterized by a function of k, δ and of the maximum number s of 1's in a row of M, and hence is polytime solvable if all three parameters are constant.
Published in the proceedings of EuroComb 2009, vol. 34 of *Electronic Notes in Discrete Mathematics*, pp. 121–125.
Keywords: consecutive-ones property, algorithm, computational complexity

1 Introduction

Let M be a binary matrix with n rows and m columns. Using the terminology of [2], we refer to a maximal sequence of consecutive ones in a row of n as a *contig*. A *gap* is a sequence of consecutive zeroes that separates two contigs; the size of a gap is the length of this sequence of zeros. M is said to have the Consecutive-Ones Property (C1P) if its columns can be permuted such that each row contains one contig (no gaps then). Such a total order of the columns of M is called a C1P-ordering of M. Deciding if a binary matrix has the C1P can be done in linear time [6]. The C1P has also been used in molecular biology, in relation with physical mapping and the reconstruction of ancestral genomes [1], where a major problem is that matrices obtained from experiments do not have the C1P [2, 1].

Handling a matrix M that does not have the C1P has been approached using different points of view. A first general approach consists of transforming M into a matrix that has the C1P, while minimizing the modifications to M; such modifications can involve either in removing rows, or columns, or both, or in flipping some entries from 0 to 1 or 1 to 0. In all cases, the corresponding optimization problems have been proven NP-hard [5, 4]. A second approach consists of relaxing the condition of consecutivity of the ones of each row, by allowing gaps, with some restriction to these gaps. The question is then to decide if there is an ordering of the columns of M that satisfies these relaxed C1P conditions. As far as we know, the only restriction that has been considered is the number of gaps, either per row or in M. It has been shown that deciding if the columns of M can be ordered in such a way that every row contains at most k contigs is NP-complete even if $k = 2$ [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 ones [3].

Here, we follow the second approach, motivated by the problem of reconstructing ancestral genomes using max-gap clusters [1]: the restrictions to the allowed gaps are that both the number of gaps per row and the size of each gap are bounded. To our knowledge, this set of restrictions has never been considered. Formally, let k and δ be two integers. M is said to have the (k,δ)-Consecutive-Ones Property, denoted by (k,δ)-C1P, if its columns can be permuted in such a way that each row contains at most k contigs and no gap larger than δ: we describe both hardness and algorithmic results on this problem.

2 Hardness Results

We show that testing for the $(2, \delta)$-C1P is NP-complete for $\delta \geq 2$ by reduction from 3SAT, building on the construction in [2]. This construction divides columns into blocks: for each variable x_i, $i = 1, \ldots, n$, we

have block b_i with two columns, and for each clause C_j, $j = 1, \ldots, m$, we have block B_j with 5 columns. Furthermore, it requires that we can force the columns into order described in Lemma 1 below. Once this order is satisfied, the remaining part of the construction, modeling each clause, remains the same as in [2], and hence, we will not repeat it here.

It is enough to show the following lemma.

Lemma 1. *There is a matrix M with columns $\cup_{i=1}^{n} b_i \bigcup \cup_{i=1}^{m} B_i$ for which in any $(2,\delta)$-C1P ordering, the columns in each block are adjacent and the blocks are ordered $b_1, \ldots, b_n, B_1, \ldots, B_n$ (or the reverse order), where $n \geq 6$.*

Proof. Let $b_i = \{b_i^1, b_i^2\}$ and $B_j = \{B_j^1, \ldots, B_j^5\}$, for every $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Let $[c_1, \ldots, c_\ell]$ denotes a row of M with ones in columns $c_1, \ldots, c_\ell$ and zeros in all other columns. If we include a block in this list, we mean all columns in this block. First, let us fix the variable blocks $b_1, \ldots, b_n$. For every $i = 1, \ldots, n$ and $j = 1, 2$, add the row $r_i^j = [b_{i-1}, b_i^j, b_{i+1}]$ to M, where $b_0 = b_{n+1} = \emptyset$. Consider $i = 3, \ldots, n-2$ and assume to the contrary that columns b_i^1 and b_i^2 are not adjacent in some $(2,\delta)$-C1P ordering O. Wlog, b_i^1 appears before b_i^2. Let $N = b_{i-2} \cup b_{i-1} \cup b_{i+1} \cup b_{i+2}$.

First, assume that there is a column $t \notin N$ between b_i^1 and b_i^2. Divide the remaining columns into 4 groups $P_1, \ldots, P_4$ such that $O = P_1, b_i^1, P_2, t, P_3, b_i^2, P_4$, i.e., for instance, P_1 is the group of columns appearing before b_i^1. Consider rows $r_{i-1}^1, r_{i-1}^2, r_{i+1}^1, r_{i+1}^2$. In each of them b_i-columns are 1, t is 0, and exactly one of columns in $b_{i-1} \cup b_{i+1}$ is 1. Hence, if two of the columns in $b_{i-1} \cup b_{i+1}$ appear in the same group, we will have at least two gaps in one of the four rows. On the other hand, if each these columns appears in a different group, then we have two gaps in the two r_i^1. Hence, all the columns between b_i^1 and b_i^2 are from the set N.

Second, assume that column b_{i-1}^1 is between b_i^1 and b_i^2. Consider again 4 groups defined by these 3 elements: $O = P_1, b_i^1, P_2, b_{i-1}^1, P_3, b_i^2, P_4$. By a similar argument as above, the columns in $A = \{b_{i-1}^2, b_{i+1}^1, b_{i+1}^2\}$ have to appear in different groups. Furthermore, columns in A cannot appear simultaneously in P_2 and P_3, or otherwise the row r_{i-1}^1 contains at least two gaps. Hence, there is one column $t_1 \in A$ in P_1 and one column $t_2 \in A$ in P_4. Now, consider the columns in b_{i-2}. The rows r_i^1 and r_i^2 contain at least one gap between a and b and placing any column in b_{i-2} between t_1 and t_2 would create another gap between t_1 and t_2. Hence, each column on b_{i-2} appears either before t_1 or after t_2. Consider again the row r_{i-1}^1, no matter whether a column in b_{i-2} is before t_1 or after t_2, it contains at least two gaps. Hence, b_{i-1}^1 is not between b_i^1 and b_i^2, and by symmetry, neither $b_{i-1}^2, b_{i+1}^1, b_{i+1}^2$ are. Using similar arguments, one can show that neither remaining elements of N can be between columns in b_i, i.e., the columns are adjacent for $i = 3, \ldots, n-2$.

It is easy to see that the blocks $b_3, \ldots, b_{n-2}$ appear in O in the correct order, and since they are at least two, b_1, b_2 must precede them and b_{n-1}, b_n must follow. Finally, for each $j = 1, \ldots, m$, to force block B_j to its right position, we add the row $[b_{n-2}, b_{n-1}^1, b_n, B_1, \ldots, B_j]$ to M.

Theorem 1. *Testing for the $(2,\delta)$-C1P is NP-complete for every $\delta \geq 2$.*

3 Algorithmic Results

A graph $G = (V, E)$ is said to have bandwidth at most b if there exists a total order on its vertices $V = \{v_1, \ldots, v_n\}$ such that every edge $\{v_i, v_j\}$ satisfies $|i - j| \leq b$. Let M be an $n \times m$ binary matrix and $G_M = (V_M, E_M)$ be the weighted graph defined as follows: $V_M = \{1, \ldots, m\}$ (each vertex of G_M represents a column of M), and there is an edge $\{i, j\}$ in E_M iff there is a row of M with entries 1 in columns i and j, and edge $e = \{i, j\}$ is weighted by the maximum of the size (number of entries 1) among all rows of M that have entries 1 in both columns i and j. The following property then follows immediately from this definition: If every row of M has at most s entries 1 and M has the (k,δ)-C1P, then G_M has bandwidth at most $s + (k-1)\delta - 1$.

In [7], Saxe describes an algorithm that decides if a graph has bandwidth at most b with complexity $O(n^{b+1})$, in time and space. We sketch now how it can be modified to test for the (k,δ)-C1P. This algorithm uses the property that, given a prefix of a total order on the vertices of a graph, if one wants to test that its bandwidth is at most b, only the b last elements of the prefix are useful; the *active region* of this prefix is then composed of its last b vertices, and it defines unambiguously the content of its prefix. The principle of the algorithm is to consider, in a breadth-first search, only the active regions, each of them defining an

equivalence class of prefixes, and given a current active region, to extend it by a vertex if it does not violate the bandwidth condition. In our problem, this algorithm needs only to be augmented by testing, each time an active region is extended, if this extension does not violate the gap conditions in any row, which adds an $O(nm)$-time cost factor to the algorithm.

Theorem 2. *Let M be an $n \times m$ binary matrix such that every row has at most s entries 1. Deciding if M has the (k,δ)-C1P can be done in time $O(nm^{s+(k-1)\delta+1})$ and space $O(m^{s+(k-1)\delta})$.*

4 Conclusion

The work we presented here leaves several questions open. The most natural is the complexity of testing for the general (k, δ)-C1P. From preliminary results that use a reduction from 3SAT but require more complicated constructions that rely on a deeper understanding of the Gapped C1P Problem, we conjecture that the general problem is NP-complete, except possibly the $(2, 1)$-C1P case, a case that remains open and is particularily interesting. It is also natural to ask if there exists a structure that can represent all orderings that satisfy some gaps conditions, as the PQ-tree does for the classical C1P. Finally, do there exist efficient (non-brute-force) algorithms for deciding the (k, δ)-C1P for small values of δ? If so, they would be practical in genomics applications.

References

1. Chauve, C. and E. Tannier, *A methodological framework for the reconstruction of contiguous regions of ancestral genomes and its application to mammalian genome*, PLoS Comput. Biol. **4** (2008), paper e1000234.
2. Goldberg, P., M. Golumbic, H. Kaplan and R. Shamir, *Four strikes againts physical mapping of DNA*, J. Comput. Biol. **2** (1995), pp. 139–152.
3. Haddadi, S., *A note on the NP-hardness of the consecutive block minimization problem*, Int. Trans. Oper. Res. **9** (2002), pp. 775–777.
4. Hajiaghayi, M. T. and Y. Ganjali, *A note on the consecutive ones submatrix problem*, Inf. Process. Lett. **83** (2002), pp. 163–166.
5. Dom, M., J. Guo and R. Niedermeier, *Approximability and parameterized complexity of the consecutive ones submatix problem*, In TAMC 2007 of Springer LNCS **4484** (2007), pp. 680–691.
6. McConnell, R., *A certifying algorithm for the consecutive-ones property*, in: *SODA* (2004), pp. 761–770.
7. Saxe, J., *Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time*, SIAM J. Algebraic Discrete Methods **1** (1980), pp. 363–369.