THE COMPUTATIONAL COMPLEXITY OF AVOIDING FORBIDDEN SUBMATRICES BY ROW DELETIONS*

SEBASTIAN WERNICKE[1], JOCHEN ALBER[2], JENS GRAMM[3],
JIONG GUO[1], and ROLF NIEDERMEIER[1]

[1] *Institut für Informatik, Friedrich-Schiller-Universität Jena,*
Ernst-Abbe-Platz 2, D-07743 Jena, Germany.
{wernicke,guo,niedermr}@minet.uni-jena.de

[2] *Digsilent GmbH,*
Heinrich-Hertz-Straße 9, D-72810 Gomaringen, Germany.
j.alber@digsilent.de

Wilhelm-Schickard-Institut für Informatik, Universität Tübingen,
Sand 13, D-72076 Tübingen, Germany.
gramm@informatik.uni-tuebingen.de

Manuscript, December 15, 2006

ABSTRACT

We initiate a systematic study of the Row Deletion(B) problem on matrices: Given an input matrix A and a fixed "forbidden submatrix" B, the task is to remove a minimum number of rows from A such that no row or column permutation of B occurs as a submatrix in the resulting matrix. An application of this problem can be found, for instance, in the construction of perfect phylogenies. Establishing a strong connection to variants of the *NP*-complete Hitting Set problem, we describe and analyze structural properties of B that make Row Deletion(B) *NP*-complete. On the positive side, the close relation with Hitting Set problems yields constant-factor polynomial-time approximation algorithms and fixed-parameter tractability results.

Keywords: Forbidden submatrix, Hitting Set, *NP*-complete, polynomial-time constant-factor approximation, fixed-parameter tractability.

1. Introduction

Forbidden subgraph problems play an important role in graph theory and algorithms (see, for instance, [2, Chapter 7]). For instance, in an application concerned with graph-modeled clustering of biological data one is interested in modifying a given graph by as few edge deletions as possible such that the resulting graph consists of a disjoint union of cliques (a so-called *cluster graph*) [19]. To solve this *NP*-complete problem, both exact exponential-time (fixed-parameter) algorithms as well as polynomial-time approximation algorithms make use of the fact that a graph is a cluster graph if and only if it contains no length-two path as a vertex-induced subgraph [3, 11, 12, 19].

*An extended abstract of this work appears in the proceedings of the *30th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'04)*, Springer, LNCS 2932, pages 349–360, held in Měřín, Czech Republic, 2004.

There is a rich literature dealing with "graph modification problems" (see, for instance, [15])—many problems here being *NP*-complete. By way of contrast, in this paper we start the so far seemingly widely neglected investigation of forbidden submatrix problems from an algorithmic point of view. Here, given an input matrix A and a fixed matrix B, the basic question is whether B is *induced* by A. This means that a *permutation* B' of B can be obtained from A by row and column deletions (where "row" means a horizontal line of a matrix, "column" a vertical line, and "permutation" that B can be transformed into B' by a finite series of row and column swappings). This work studies corresponding "matrix modification problems" where, given A and a fixed B, we are asked to remove as few rows from A as possible such that the resulting matrix no longer induces B.

Whereas there has been some research into randomized testing algorithms for the occurrence of forbidden submatrices [8], the corresponding problem of removing these matrices has seemingly not yet been studied despite some practical relevance: For instance, they appear in computational biology in the context of constructing "perfect phylogenies" [18, 20]. Here, a binary input matrix A allows for a perfect phylogeny if and only if A does not induce the submatrix B consisting of the rows $(\,1\;1\,)$, $(\,1\;0\,)$, and $(\,0\;1\,)$ (see [18, 20] for details).

This work initiates a systematic study of matrix modification problems concerning the complexity of row deletion for forbidden submatrices. Our main result is to establish a very close link between many of these problems and restricted versions of the *NP*-complete Hitting Set problem [10]. We describe and analyze structures of the forbidden submatrix B which make the corresponding row deletion problem "equivalent" to particular versions of Hitting Set. On the negative side, this implies *NP*-completeness for most row deletion matrix modification problems, holding already for a binary alphabet. On the positive side, we can also show that polynomial-time approximation and fixed-parameter tractability results for Hitting Set carry over to the corresponding row deletion problems.

To the best of our knowledge, no such systematic study has been undertaken so far. We are only aware of two related works: First, Klinz, Rudolf, and Woeginger [13] deal with the permutation of matrices (without considering row deletions) in order to avoid forbidden submatrices. There, however, the authors consider the case of permuting rows and columns of the input matrix A to obtain a matrix A' such that A' cannot be transformed into a fixed matrix B by row and column deletions. Among other things, they show *NP*-completeness for the general decision problem. Second, Damaschke [4] studies, among others, parameterized algorithms for the problem of *enumerating* all ways of deleting at most k rows or columns from a given input matrix such that the resulting matrix does not induce the forbidden submatrix that consists of the rows $(\,1\;1\,)$, $(\,1\;0\,)$, $(\,0\;1\,)$, and $(\,0\;0\,)$.

Our work is structured as follows. In Section 2, we start with basic definitions and some easy observations. Then, in Section 3 we show how Row Deletion(B) can be solved using algorithms for Hitting Set problems, using a "parameter-preserving reduction" (the core tool of this paper). After that, in Section 4, the main results of the work are presented, giving several parameter-preserving reductions from Hitting Set problems to Row Deletion(B) for various types of the forbidden submatrix B. Finally, we end with some concluding remarks and open problems in Section 5.

2. Definitions and Preliminaries

All matrices in this work have entries from an alphabet Σ of fixed size ℓ; we call these matrices *ℓ-ary*. Note, however, that all computational hardness results already hold for binary alphabets. The central problem Row Deletion(B) for a fixed matrix B is defined as follows.

Row Deletion(B)
Input: A matrix A and a nonnegative integer k.
Question: Using at most k row deletions, can A be transformed into a matrix A' such that A' does not induce B?

Herein, a matrix A is said to *induce* B if we can select a subset of the rows and columns in A such that the resulting submatrix—an *occurrence* of B in A—is a *permutation* of B. By permutation we mean a matrix that can be obtained from B through a finite series of row and column swappings. A matrix A is *B-free* if there is no occurrence of B in A.

This work establishes strong links between Row Deletion(B) and d-Hitting Set for constant d, which is defined as follows.

d-Hitting Set
Input: A finite set S, a collection $\mathcal{C}$ of subsets of S where each subset has size at most d, and a nonnegative integer k.
Question: Is there a subset $S' \subseteq S$ with $|S'| \leq k$ such that S' contains at least one element from each subset in $\mathcal{C}$?

Already for $d = 2$, d-Hitting Set is *NP*-complete [10].

To express the closeness between variants of Row Deletion(B) and d-Hitting Set for various d, we need the following strong notion of reducibility. Let $(S, \mathcal{C}, k)$ be an instance of d-Hitting Set. We say that there is a *parameter-preserving reduction* from d-Hitting Set to Row Deletion(B) if there is a polynomial-time algorithm that transforms $(S, \mathcal{C})$ into a matrix A such that $(S, \mathcal{C}, k)$ is a true-instance of d-Hitting Set if and only if (A, k) is a true-instance of Row Deletion(B). The important observation here is that the "objective value parameter" k remains unchanged. This makes it possible to directly link polynomial-time approximation as well as exact (fixed-parameter) algorithms for both problems.

In the next section, we show how fixed-parameter algorithms and polynomial-time approximation algorithms for Hitting Set can also be used for Row Deletion(B). As to the terminology (deferring a more detailed introduction to [5, 9, 16]), a problem is called *fixed-parameter tractable* if it can be solved in $f(k) \cdot n^{O(1)}$ time, where n is the size of the input (as in classical complexity theory), k is a *parameter* (usually a positive integer) specified by the input, and f is a computable function only depending on k. We use the term *polynomial-time approximation algorithm* to refer to an algorithm that, in polynomial-time, computes a solution to a given optimization problem such that its cost is guaranteed to be within a certain range of that of an optimal solution (see, for instance, [1, 6, 21] for more details on the topic of approximation algorithms).

3. Algorithmic Tractability

This section points out an algorithmic approach to solve Row Deletion(B). To this end, we give a parameterized reduction to r-Hitting Set where r is the number of rows in B.

In order to actually perform row deletions in the input matrix A of Row Deletion(B) it is of course necessary to find the set of rows in A that induce B. A straightforward algorithm yields the following.

Proposition 1 *Given an $n \times m$ matrix A and a fixed $r \times s$ matrix B (where $1 \leq r \leq n$ and $1 \leq s \leq m$), we can find all size-r sets of rows in A that induce B in $O(n^r \cdot m \cdot s \cdot r)$ worst-case time.*

Proof. Using an exhaustive approach, we try all $\binom{n}{r}$ ways to select a size-r subset of rows in A and all $r!$ ways to permute this selection. For each of the resulting $\binom{n}{r} \cdot r! = O(n^r)$ permuted submatrices of A, it is possible to check in $O(m \cdot s \cdot r)$ time whether s columns can be selected in it that match all s columns of B one-to-one. □

Observe that Proposition 1 gives a pure worst-case estimation not making use of any special structure the respective forbidden submatrix B might have. In any case, however, for constant values of r and s the running time is polynomial.

Theorem 2 *Given a fixed $r \times s$ submatrix B,* Row Deletion(B) *is parameter-preserving reducible to r-*Hitting Set *in $O(n^r \cdot m \cdot s \cdot r)$ time. (The parameter is the number k of allowed row deletions.)*

Proof. Given an instance (A, k) of Row Deletion(B) (where A is an $n \times m$ matrix), we can construct an instance $(S, \mathcal{C}, k)$ of r-Hitting Set as follows:

1. Let the set S contain one element for every row in A.
2. Starting with $\mathcal{C} = \emptyset$, compute a collection $\mathcal{C}$ of subsets of S as follows: For every size-r subset $R \subseteq S$ that corresponds to r rows in A which induce B, add R to $\mathcal{C}$.
3. The parameter k is directly preserved.

The claimed running time follows with Proposition 1. An example for the reduction is given in Figure 1.

The 1:1-correspondence between solutions of the r-Hitting Set instance and the Row Deletion(B) instance is rather obvious, but we include it here for sake of completeness and as a preparation for the more complex arguments in Section 4. Let $\mathcal{S}' \subseteq \mathcal{S}$ be a solution of size k to the r-Hitting Set instance $(S, \mathcal{C}, k)$. We

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
r_1	0	1	1	0	0	0	1	0	0	0
r_2	0	0	0	0	0	0	0	0	0	1
r_3	0	0	0	1	0	0	0	0	1	1
r_4	0	0	0	0	0	0	1	1	1	1
r_5	0	0	0	0	0	0	1	1	0	0
r_6	0	1	0	0	1	1	0	0	0	0

$$\mathcal{C} = \{\{r_1, r_2, r_4\}, \{r_1, r_3, r_4\}, \{r_1, r_4, r_6\}, \{r_1, r_5, r_6\}, \{r_2, r_4, r_5\}, \{r_3, r_4, r_5\}\}$$

Figure 1: Illustrating the reduction from ROW DELETION(B) to r-HITTING SET. Let the forbidden submatrix B consist of the rows $(1\,1)$, $(0\,1)$, and $(1\,0)$. The grey underlay shows, as an example, how B is induced by rows r_1, r_4, and r_6 in columns c_2 and c_7 of an input matrix A, which causes the subset $\{r_1, r_4, r_6\}$ to be added to $\mathcal{C}$.

delete the rows in A that correspond to the elements in $\mathcal{S}'$, yielding A'. Assume that B were still induced in A' by a set I of rows. Then, the rows in I induce B in the original input matrix A and hence—by means of construction—a set containing the elements corresponding to these rows is contained in $\mathcal{C}$. But one row of I must then have been deleted since $\mathcal{S}'$ is a valid solution to $(S, \mathcal{C}, k)$, a contradiction. Therefore, B cannot be induced by A' anymore.

If, by way of contrast, A can be made B-free by deleting k rows then, for each occurrence of B in A, at least one of the rows responsible for the occurrence must have been deleted. By choosing the elements corresponding to the deleted rows as a solution $\mathcal{S}' \subseteq \mathcal{S}$ to the generated r-HITTING SET instance, we have chosen at least one element from every set in $\mathcal{C}$, making $\mathcal{S}'$ a valid solution of size k. □

Theorem 2 directly implies the following three positive results:

1. The linear-time approximation algorithm for r-HITTING SET with approximation factor r [6] can be used to obtain a factor-r approximation for ROW DELETION(B).

2. A straightforward algorithm solves r-HITTING SET in $O(r^k \cdot n)$ time, where k denotes the size of the solution. This means that for constant r, ROW DELETION(B) is fixed-parameter tractable with respect to the parameter k. (However, note that the $O(n^r)$ factor in the running time of the reduction in Theorem 2 means that whereas r-HITTING SET is fixed-parameter tractable when parameterized by both r and k, our result does not show that this holds for ROW DELETION(B). Since a matrix can contain up to $O(\binom{n}{r})$ copies of the forbidden submatrix B, it is conceivable that ROW DELETION(B) is not fixed-parameter tractable when parameterized by both r and k.)

3. Faster algorithms for r-HITTING SET directly yield faster algorithms for ROW DELETION(B); see [7, 17] for the currently best fixed-parameter algorithms for r-HITTING SET (for instance, the best exponential term for 3-HITTING SET is known to be 2.18^k instead of the straightforward to achieve exponent 3^k).

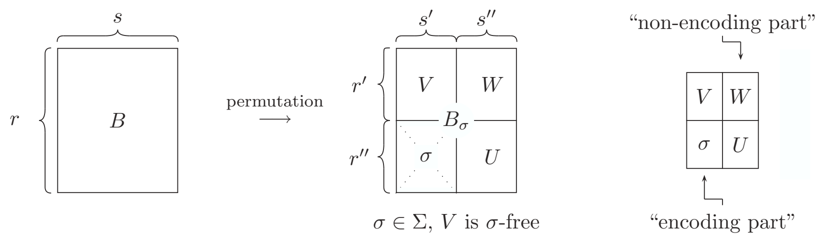

Figure 2: General scheme for the σ-decomposition of a matrix B over the alphabet Σ.

4. Computational Hardness

In this section, we explore the relative computational hardness of Row Deletion(B) by studying its relationship to d-Hitting Set. We point out many cases concerning the structure of the forbidden submatrix B for which solving Row Deletion(B) is at least as hard as d-Hitting Set for some d depending only on B.

4.1. Summary of Results

The key idea behind all reductions from d-Hitting Set to Row Deletion(B) is to choose a symbol σ from the alphabet Σ and to decompose B—in a certain manner—into four submatrices, one of which consists only of σs and another one of which does not contain σ. We can then use the part of the decomposition that does not contain σ to encode a given d-Hitting Set instance into an instance of Row Deletion(B)—that is, a matrix A—and rely on σ as a "fill-in" symbol for those entries in A that are not touched by the encoding.

We call this special decomposition of the forbidden submatrix B a *σ-decomposition*, illustrated in Figure 2, and formally define it as follows:

Definition 3 (σ-decomposition) *Given an ℓ-ary $r \times s$ matrix B over an alphabet Σ, a permutation $B_\sigma = (b_{ij})$ of B is called a* σ-decomposition *of B if there exist nonnegative integers r', r'', s', s'' with $r' + r'' = r$, $s' + s'' = s$ and a letter $\sigma \in \Sigma$ such that*

1. *$r' > 0$ and $s' > 0$,*
2. *$\forall\ 1 \leq i \leq r', 1 \leq j \leq s' :\ b_{ij} \neq \sigma$ (call this upper left submatrix V) and*
3. *$\forall\ r' < i \leq r, 1 \leq j \leq s' :\ b_{ij} = \sigma$.*

The upper right $r' \times s''$ submatrix $(b_{ij})_{1 \leq i \leq r', s' < j \leq s}$ of B_σ is denoted by W, the lower right $r'' \times s''$ submatrix $(b_{ij})_{r' < i \leq r, s' < j \leq s}$ is referred to as U. The left part of B_σ (the one containing V) $(b_{ij})_{1 \leq i \leq r, 1 \leq j \leq s'}$ is called the encoding part *of B_σ. The right part $(b_{ij})_{1 \leq i \leq r, s' < j \leq s}$ of B_σ (the one consisting of W and U) is called the* non-encoding part *of B_σ.*

For a given $\sigma \in \Sigma$, a corresponding σ-decomposition can easily be computed in time only linearly depending on the size of B: First, select a subset of columns in B that have the property that, for each row, either every column has an entry equal to σ or no column has an entry equal to σ. Then permute the rows in B such that all σs in the selected columns are moved to the bottom of B. Finally, permute the columns such that the selected columns are all moved to the left. This resulting matrix is clearly a σ-decomposition. Whereas we assume a given σ-decomposition for the forbidden submatrix in our main theorems, its computation appears to be somewhat of interest in its own right and can serve to test whether a given matrix fits the properties of some of the main theorems.

In the following hardness results, the height r' of V plays a crucial role. For better readability, we only present the results here, deferring their proofs to Section 4.2.

Theorem 4 *Let B be a forbidden submatrix of size $r \times s$ with a σ-decomposition B_σ where the submatrix V (of height r') of B_σ is not induced in the non-encoding part of B_σ. Then there exists a parameter-preserving reduction from r'-*Hitting Set *to* Row Deletion*(B). In particular, if $r' \geq 2$,* Row Deletion*(B) is NP-complete.*

It is of course possible that V *is* induced in the non-encoding part of B_σ. In that case, each column vector of V is induced at least once in the non-encoding part. If we can find one column vector of V which is induced *at most once* in the non-encoding part, we are again able to achieve a hardness result for Row Deletion(B):

Theorem 5 *If the $r \times s$ matrix B has a σ-decomposition B_σ where the submatrix V of height r' has a column vector that is induced at most once in the non-encoding part of B_σ, then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

Observe that it is possible to construct a matrix B which fulfills the prerequisites of Theorem 4 but does not fulfill the prerequisites of Theorem 5, and vice versa. For example, the matrix $\left(\begin{smallmatrix} 1&1&1&0&0\\ 1&1&1&0&0\\ 0&0&0&1&1\\ 0&0&0&1&1 \end{smallmatrix}\right)$ over the binary alphabet $\Sigma = \{0, 1\}$ fulfills the prerequisites of Theorem 4 but not of Theorem 5. The converse holds, for instance, for the matrix $\left(\begin{smallmatrix} 1&1\\ 1&0\\ 0&1 \end{smallmatrix}\right)$.

If the submatrix B does not fulfill any of the two prerequisites from Theorems 4 or 5, we are able to determine two further subcases for which a hardness result can be established:

Theorem 6 *Let B be a forbidden $r \times s$ submatrix with a σ-decomposition B_σ where all entries of the submatrix U are equal to σ and V contains r' rows. Then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

Theorem 7 *Let B be a forbidden $r \times s$ submatrix with a σ-decomposition B_σ where all entries of the submatrix W are equal to σ and V contains r' rows. Then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

For all other cases, that is, if B does not fulfill any of the prerequisites from Theorems 4–7, we are not aware of a general statement on the complexity of Row Deletion(B). As an example consider the matrix $B = \left(\begin{smallmatrix}1&0&0\\0&1&0\\0&0&1\end{smallmatrix}\right)$ over the alphabet $\Sigma = \{0, 1\}$. This matrix only fulfills the prerequisites of Theorem 7 with $V = (1)$, implying that the corresponding Row Deletion(B) problem is at least as hard to solve as 1-Hitting Set. However, this is not a very useful lower complexity bound since 1-Hitting Set is polynomial-time solvable whereas Row Deletion(B) with $B = \left(\begin{smallmatrix}1&0&0\\0&1&0\\0&0&1\end{smallmatrix}\right)$ is *NP*-complete.[a]

It is also clear that Row Deletion(B) is not *NP*-hard for all B. For example, Row Deletion(B) is always solvable in polynomial time if B is a 1×1-matrix (every row that contains B has to be removed). There are also larger matrices B for which Row Deletion(B) is solvable in polynomial time, for instance, the matrix $B = \binom{1}{0}$ over the alphabet $\Sigma = \{0, 1\}$: Observe that a B-free matrix A has the property that all of its columns consist either solely of 1s or solely of 0s, that is, all rows of A need to be identical. This implies that the minimum number of rows that need to be deleted in order to make an $n \times m$ matrix B-free is equal to $n - x$, where x denotes the size of the largest set of identical rows in A, which can be determined in linear time (with respect to the size of A).

Note, however, that Row Deletion(B) for a forbidden submatrix $B = \binom{1}{1}$ over the alphabet $\Sigma = \{0, 1\}$ already is *NP*-complete by Theorem 4, since in that case B trivially has a σ-decomposition with $\sigma = 0$ and $V = B$.

4.2. Hardness Proofs

As mentioned above, all hardness results are proven by encoding an instance $(S, \mathcal{C}, k)$ of d-Hitting Set—where the value of d is determined by the forbidden submatrix B—as an instance (A, k) of Row Deletion(B). The key idea as to how we can use a given σ-decomposition of the forbidden submatrix B to encode a d-Hitting Set instance is illustrated by the following proof of Theorem 6. Subsequently, we indicate how this construction can be extended to prove, in ascending involvedness of the construction, Theorem 7, Theorem 4, and Theorem 5.

Proof. [Theorem 6] Given an instance $(S, \mathcal{C}, k)$ of r'-Hitting Set and a σ-decomposition B_σ of the forbidden $r \times s$ submatrix B, let $S = \{1, \ldots, n\}$ and $\mathcal{C} = \{C_1, \ldots, C_m\}$. For now, assume that $r' = r$, that is, $r'' = 0$ and B consists only of V and W (the case $r' < r$ is dealt with at the end of this proof). We generate a matrix A of size $n \times (s \cdot m)$. Each row of A corresponds to an element in S. For each set $C \in \mathcal{C}$, one occurrence of B is encoded into s consecutive columns of A, thereby

[a] The *NP*-hardness can be shown as follows: Given a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$, construct a binary $n \times m$ matrix $A = (a_{ij})$ where for each edge $e_k = \{v_i, v_j\} \in E$, $a_{ik} = 0$ and $a_{jk} = 0$; all other entries of A are equal to 1. Observe how each triangle contained in G causes the forbidden submatrix $B = \left(\begin{smallmatrix}1&0&0\\0&1&0\\0&0&1\end{smallmatrix}\right)$ to be induced in A. Hence, solving Row Deletion(B) on A is equivalent to asking for the minimum number of vertices in G that need to be deleted in order to make the resulting graph triangle-free, an *NP*-complete problem [14].

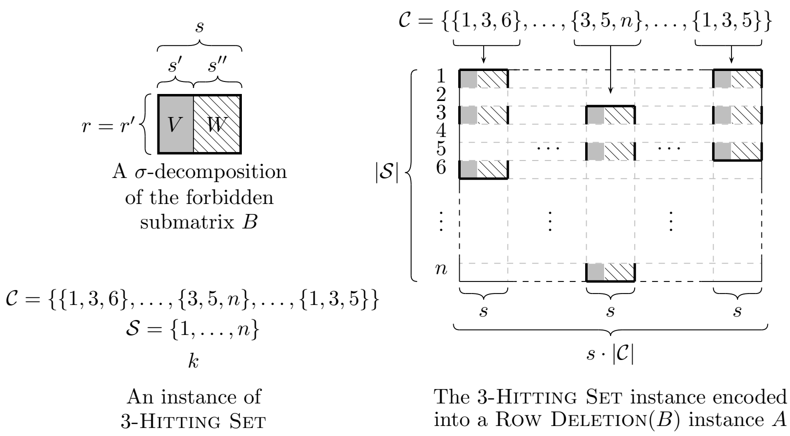

Figure 3: An example reduction from 3-HITTING SET to ROW DELETION(B) following from the proof of Theorem 6 (illustrated for the case where $r' = r$).

using only the rows that correspond to the elements in C. For example, consider $C_h \in \mathcal{C}$, $1 \leq h \leq m$, with $C_h = \{z_1, \ldots, z_r\}$ and $z_1, \ldots, z_r \in \{1, \ldots, n\}$. Then we generate the submatrix $(a_{ij})_{1 \leq i \leq n, (h-1) \cdot s < j \leq h \cdot s}$ of A such that the z_ith row of this submatrix equals the ith row of B, for all $i = 1, \ldots, r$, and all other rows of this submatrix are set equal to σ (an illustration for this is provided in Figure 3). In this way, A contains m *blocks* of s consecutive columns where each block induces B exactly once. These are the only occurrences of B, since for every set of r rows it holds that two columns from two different blocks contain at least one row with a σ. The outlined reduction can be performed in $O(n \cdot m \cdot s)$ time.

The construction generalizes to the case where $r' < r$ (that is, where $r'' > 0$) by adding $r'' + k$ rows containing only σ to the bottom of A. This addition achieves two things: First, since U also consists only of σ entries, each block in A into which the upper r' rows of B have been written then automatically induces B. Second, it is not possible to delete at most k of the added rows such that an occurence of B is destroyed. Note that by this construction, although B is induced multiple times in each block, V is induced only once in each block.

Solutions to the original instance of r'-HITTING SET have a 1:1-correspondence with solutions to ROW DELETION(B) on A:

($\Rightarrow$) Assume that we have a solution $\mathcal{S}'$ to the original instance of r'-HITTING SET with $|S'| = k$. Delete those rows in A that correspond to the elements in $\mathcal{S}'$, obtaining A'. Note that then, from the submatrix V of each B that was encoded into A, at least one row has been deleted. Every column in A' contains less than r' symbols different from σ. This directly implies that V does not occur in A', and therefore A' is B-free, that is, we have a solution for the ROW DELETION(B) instance with k row deletions.

($\Leftarrow$) Assume that by deleting at most k rows in the input matrix A we can make it B-free. Note that we cannot destroy any of the induced Bs in A by deleting at most k of the bottom $r'' + k$ rows of A. Therefore, it must be possible to delete at most k of the top n rows of the matrix A to make it B-free. Furthermore, from each induced B in A, at least one row must have been deleted since A would otherwise not be B-free. Thus, choosing the elements in S that correspond to the deleted rows into a set $\mathcal{S}'$ yields a solution of size k to the original r'-HITTING SET problem. □

The idea of the above proof—using the submatrix V of B_σ to encode an instance of d-HITTING SET—is employed in all of the following proofs. In order to show the 1:1-correspondence of the original d-HITTING SET problem and the generated ROW DELETION(B) instance, mainly two conditions need to be fulfilled:

(i) If the optimal solution to the d-HITTING SET instance has size k, there are no "cheaper" solutions for the generated ROW DELETION(B) instance.

(ii) If there is a solution of size k to the original d-HITTING SET instance, deleting the corresponding rows in A destroys all occurrences of B in A.

Whilst Condition (i) is rather straightforward to meet by extending the idea of the above proof, Condition (ii) is quite intricate to fulfill in general, because it must be ensured that the parts of the submatrices B that are encoded into A due to the sets in $\mathcal{C}$ do not induce additional unwanted occurrences of B. For example, the prerequisites of Theorem 7, which we are about to prove next, allow the submatrix U in the σ-decomposition to induce V, and hence care needs to be taken that this does not lead to unwanted occurrences of B in the construction.

Proof. [Theorem 7] Given an instance of r'-HITTING SET and a σ-decomposition for the forbidden submatrix B, the resulting matrix A of this proof's reduction is composed of four submatrices: The upper left submatrix is generated by the encoding scheme presented in the proof of Theorem 6 using V as the forbidden submatrix, the upper right and lower left submatrices are filled with σs. (In the following, we will sometimes use this notion by referring to the *left* and *right part* of the constructed matrix A.) Two cases are distinguished for placing U into the lower right submatrix of A:

In Case 1 where U does not induce V, the lower right submatrix has size $(k+1)r'' \times (k+1)s''$ and contains $k+1$ times the matrix U in a diagonal scheme. In Case 2 where U *does* induce V, the lower right submatrix has size $(k+1)r'' \times s''$ and contains $k+1$ copies of U. The two cases are illustrated in Figure 4.

Observe that the reduction for Case 1 keeps the right part of A V-free. Recall that a single U by itself cannot induce V according to the prerequisite of Case 1. However, if we would encode occurrences of U one upon the other (as for Case 2) then, on the one hand, V could be induced by rows from different encodings of U in the lower right part of A. On the other hand, U could be induced by several encodings of V in the upper left part of A. Consequently, we would have unwanted occurrences of B in A. The diagonal scheme avoids these unwanted occurrences of V and, in particular, yields that the right part of A is B-free. In Case 2, the reduction cannot keep the right part of A V-free because already a single occurrence of U induces V. However, B cannot be induced there because the right part of A contains only $s'' < s$ columns (that is, less columns than B has). Therefore, if we

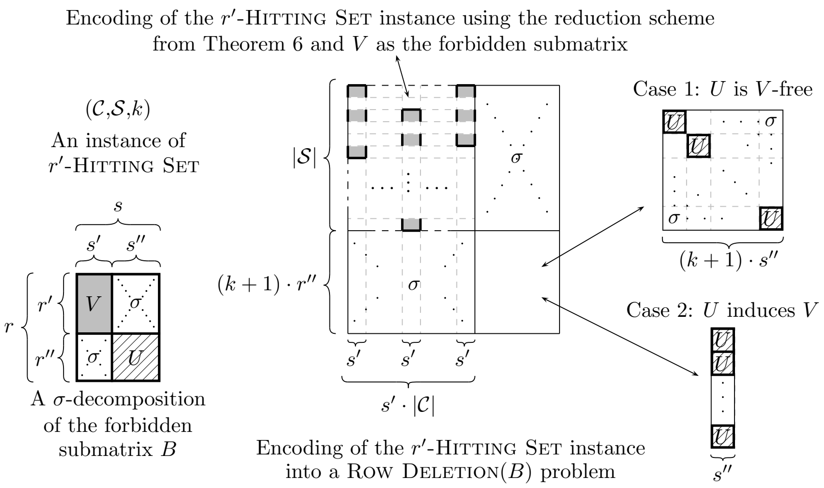

Figure 4: Reduction from r'-Hitting Set to Row Deletion(B) used in the proof of Theorem 7.

destroy all induced Vs in the upper left part of A, then U cannot be induced in the left part of A due to the prerequisite of Case 2 that U induces V. Then, the matrix A can be made B-free, even if there are some occurrences of V in the right part of A. (Note that by the shape of A, there is no combination of columns in the left and right part of A that together induce V.)

Now we show the 1:1-correspondence of solutions:

($\Rightarrow$) Assume that we have a solution $\mathcal{S}'$ to the encoded r'-Hitting Set instance. Then, delete those of the n topmost rows in A that correspond to the elements in $\mathcal{S}'$, obtaining A'. Note that this destroys all occurrences of V in the left part of A, thereby ensuring Condition (ii). For Case 1, this directly implies that A' is V-free and therefore B-free. For Case 2, recall that B is not induced in the right part of A since we can find no sufficiently large set of columns such that both U and V are induced in the corresponding submatrix. Since V is not induced in the left part of A', this also means that U is not induced in this submatrix. Hence, A' is B-free.

($\Leftarrow$) Assume that by deleting k rows, we can make A B-free. Note that by deleting one of the $(k+1) \cdot r''$ bottommost rows in A, we can destroy at most one induced B in A. This implicates that there is an optimal solution to Row Deletion(B) which involves only the deletion of k of the n topmost rows of A. The rest of the argument follows from the proof of Theorem 6: If, by deleting k of the n topmost rows of A, we can make A B-free, then from each V encoded into the left part of A, at least one row must have been deleted, ensuring Condition (i). Since each encoded V corresponds directly to a set in $\mathcal{C}$, choosing the elements in S that correspond to the deleted rows in A yields a solution to the original r'-Hitting Set instance. □

We now turn our attention to the case where we have a σ-decomposition B_σ for the forbidden submatrix in which neither W nor U consists solely of σs. The only

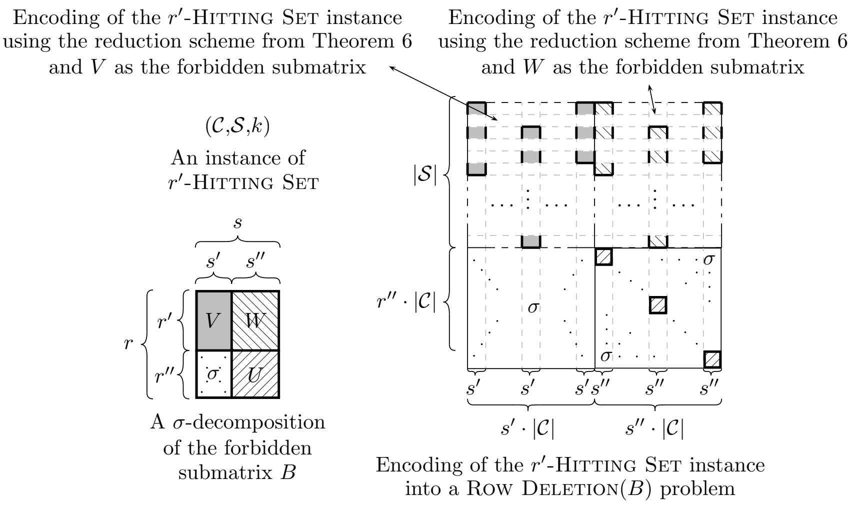

Figure 5: Reduction from r'-Hitting Set to Row Deletion(B) used in the proof of Theorem 4.

prerequisite is that V is not induced the non-encoding part of B_σ.

Proof. [Theorem 4] As illustrated in Figure 5, this reduction from a given instance $(S, \mathcal{C}, k)$ of r'-Hitting Set to an instance (A, k) of Row Deletion(B) is similar to the one used in the proof for Case 1 of Theorem 7. The resulting matrix A is again composed of four submatrices, the reduction only differs in the construction of the upper right submatrix of A. In the upper right submatrix of A, the given r'-Hitting Set instance is encoded using the scheme from the proof of Theorem 6 and W as the forbidden submatrix. As in the previous two proofs, the parameter k is preserved. Also, it is clear that the encoding can be carried out in polynomial time with respect to the input size.

As in the proof of Theorem 7, the encoding process ensures that B is not induced in the right part of A. This is due to the following observation: Assume that B is induced in the right part of A. Then V is induced there as well. By the prerequisites of the theorem, the non-encoding part of B does not induce V. Therefore, an occurrence of V involves columns from at least two different encodings of U. However, note that any two such columns of A can only have less than r' rows that do not contain a σ: No two W are encoded into the same set of rows and the U are encoded in a diagonal scheme. Hence, because V has r' rows that do not contain a σ, we have a contradiction.

Now observe that by deleting one of the $r'' \cdot |\mathcal{C}|$ bottom rows in A, we can destroy at most one induced B in A. This could also be achieved by deleting one of the n topmost rows of A. Therefore, the given solution can be translated into an optimal solution that only deletes some of the topmost n rows. Then, using the same argument as in the previous proofs, the elements of S that correspond to the deleted rows form a solution for the original r'-Hitting Set instance. Conversely,

$$B_\sigma = \begin{bmatrix} 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{array}{l} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} i_{11} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} i_{10} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} i_{01} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} i_{00} \end{array} \begin{array}{l} \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\} r' \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\} r'' \end{array} \qquad V = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad W = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \sigma = 0 \quad U = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Figure 6: Sorted decomposition of a binary matrix B according to the proof of Lemma 8. To the right of B, the decomposition submatrices V, W, and U are shown.

if we have a solution of size at most k to the original r'-Hitting Set instance, deleting the corresponding rows in A destroys all occurrences of V in A making it B-free. Hence, every solution to the r'-Hitting Set instance $(S, \mathcal{C}, k)$ implies a solution to the Row Deletion(B) instance (A, k) and vice versa. □

The scheme and key ideas of the above proof are also used in proving Theorem 5. The details of the proof are rather involved, however. We first establish a result for binary $r \times 2$ matrices in Lemma 8, which is then extended to larger alphabets in Lemma 9. This lemma is then used to obtain a proof for Theorem 5.

Lemma 8 *Let B be a binary $r \times 2$ matrix. If, for a $\sigma \in \Sigma = \{0, 1\}$, B contains a column with r' symbols different from σ, then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

Proof. We will assume without loss of generality that we are given a *maximal* σ-decomposition B_σ of B. This means that it is not possible to decompose B such that the decomposition-submatrix V has more rows, even when choosing another $\sigma \in \Sigma$. For the rest of this proof, we let $\sigma = 0$ for the sake of readability.

We can assume that the non-encoding part of B_σ is not empty; otherwise, the Lemma is true by Theorem 4. Thus, since B_σ is only of width two, the decomposition-matrices V, W, and U are all column vectors and the following observations can be made for B:

- None of the two columns in B may contain 0 more than r' times: Otherwise, choosing $\sigma = 1$ leads to a submatrix V with more rows in the corresponding decomposition of B—a contradiction to our assumption that the given decomposition is maximal.
- The right column may not contain more 1s than the left column in the decomposition because then the right part of B_σ would be the encoding part in a maximal σ-decomposition of B.

The second observation leaves us with two cases to consider for the given σ-decomposition of B: Either, the right column contains less than r' or exactly r' 1s.

The first case is already handled by Theorem 4, because if the second column in the decomposition contains less than r' 1s, then it cannot induce V since V contains exactly r' 1s. Therefore, we only need to consider the case where B is a binary $r \times 2$ matrix and each column contains exactly r' entries equal to 1. Furthermore, we assume that B is sorted such that—from top to bottom—the first i_{11} row vectors in B are equal to $(1\,1)$, the next i_{10} row vectors are equal to $(1\,0)$, the next i_{01} row vectors are equal to $(0\,1)$, and the bottom i_{00} row vectors are equal to $(0\,0)$. The resulting matrix B_σ and the submatrices V, W, and U of such a sorted decomposition are shown in Figure 6.

Note that this is a maximal decomposition of B where $i_{11} + i_{10} = r'$ and—since each column contains exactly r' entries equal to 1—where $i_{10} = i_{01}$. Additionally, we can assume that $i_{11} > 0$, because the case $i_{11} = 0$ is already included in Case 2 in the proof of Theorem 7. With these assumptions in place, we construct the output matrix A that is a "parameter-equivalent" instance of Row Deletion(B) for a given instance $(S, \mathcal{C}, k)$ of r'-Hitting Set where $S = \{1, \ldots, n\}$ in two steps:

1. Use the construction employed in the proof of Theorem 4 to obtain a matrix $\tilde{A}$ from $(S, \mathcal{C}, k)$ and B.
2. Exhaustively perform the following "merge-operation" on $\tilde{A}$: While there are two columns that induce B in the right part of $\tilde{A}$ (that is, the part where U and W are encoded), arbitrarily choose one of these two columns and delete it. Call the resulting matrix A.

Performing the merge-operation is justified as follows: Let c_1 and c_2 be two columns in the left part of $\tilde{A}$ and c'_1 and c'_2 be two columns in the right part of $\tilde{A}$ such that c_1 induces B with c'_1 and c_2 induces B with c'_2. If, additionally, c'_1 and c'_2 induce B, they have to induce the row vector $(1\,1)$ exactly i_{11} times. Due to the construction of $\tilde{A}$, this row vector can only be induced in the upper part of $\tilde{A}$. Note that, additionally, each column in the upper right part of $\tilde{A}$ contains exactly i_{11} 1s. Hence, if c'_1 and c'_2 induce the row vector $(1\,1)$ exactly i_{11} times, their upper n entries must be identical. But then observe how c_1 also induces B together with c'_2 and c_2 induces B together with c'_1. Therefore, we can make the observation that after the merge-operation, both c_1 and c_2 still induce B with some column in the right part of $\tilde{A}$, that is, the merge operation only destroys redundant occurrences of the forbidden submatrix.

We now claim that Row Deletion(B) has a solution of size k for A if and only if the encoded r'-Hitting Set instance has a solution of size k. To prove this claim, we first show that there always exists an optimal solution to Row Deletion(B) on A that does not involve the deletion of any of the bottom $r'' \cdot |\mathcal{C}|$ rows. In order to see this, note that the following holds true after the merge-operation: If a set M of columns was merged to a single column $c \in M$, then c contains r' 1s, some of which are to be found in the top n rows of A. Therefore, a B induced by c and some column in the left part of A which can be destroyed by deleting one of the bottom $r'' \cdot |\mathcal{C}|$ rows of A can also be destroyed by deleting one of those top n rows of A in which c contains a 1.

Having thus established that there always exists an optimal solution to Row Deletion(B) on A that only involves the deletion of k of the top n rows, the same arguments as in the proof of Theorem 4 can be employed to show the 1:1-correspondence of solutions between the original r'-Hitting Set instance and Row Deletion(B) on A. □

Let us illustrate the reduction in the above proof by the following concrete example: Let $B = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ be the forbidden submatrix. This is already a maximal σ-decomposition with $V = \left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$ and $\sigma = 0$. For the reduction from 2-Hitting Set to Row Deletion(B) in this example, we shall use an instance of 2-Hitting Set where $S = \{1, 2, 3, 4, 5\}$, $\mathcal{C} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 5\}, \{3, 4\}\}$, and $k = 3$. Then $\tilde{A}$ is generated by the reduction scheme from the proof of Theorem 4, yielding

$$\tilde{A} = \begin{pmatrix} 1\,1\,1\,0\,0\,0 & 1\,1\,1\,0\,0\,0 \\ 1\,0\,0\,1\,1\,0 & 0\,0\,0\,1\,1\,0 \\ 0\,1\,0\,1\,0\,1 & 0\,0\,0\,0\,0\,1 \\ 0\,0\,1\,0\,0\,1 & 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 & 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,0\,0\,0\,1 \end{pmatrix}.$$

After performing the merge-operation (the 7th, 8th and 9th column are merged and the 10th and 11th column are merged), we have

$$A = \begin{pmatrix} 1\,1\,1\,0\,0\,0 & 1\,0\,0 \\ 1\,0\,0\,1\,1\,0 & 0\,1\,0 \\ 0\,1\,0\,1\,0\,1 & 0\,0\,1 \\ 0\,0\,1\,0\,0\,1 & 0\,0\,0 \\ 0\,0\,0\,0\,1\,0 & 0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 1\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,1\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,0 \\ 0\,0\,0\,0\,0\,0 & 0\,0\,1 \end{pmatrix}.$$

Note how after the merge-operation, each one of the six left columns induces the forbidden submatrix with at least one of the three rightmost columns and that the three rightmost columns do not induce B by themselves.

We can now generalize Lemma 8 to alphabets of arbitrary size:

Lemma 9 *Let B be an ℓ-ary $r \times 2$ matrix over the alphabet Σ. If, for a $\sigma \in \Sigma$, B contains a column with d symbols different from σ, d-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

Proof. We only need to prove the lemma for the case where the left column of B is induced in its right column and its right column is induced in its left column. If this is not the case, the lemma is already proven by Theorem 4.

To prove the lemma, we use the same construction as in the proof of Lemma 8, again employing the merge-operation. The correctness of the merge-operations for an alphabet of size greater than two is justified as follows: In the proof of Lemma 8, we considered the i_{11} rows of B which induced the row vector $(\,1\;1\,)$. Now, given a maximal σ-decomposition B_σ of a forbidden submatrix of size $r \times 2$, let i be the number of rows in B_σ that do not contain σ. Then for the matrix $\tilde{A}$, if B is induced by two columns c_1 and c_2 in the right part of $\tilde{A}$, the upper n rows of these columns must induce i row vectors that do not contain σ. But then the top n rows of the columns c_1 and c_2 must be identical because these rows contain only i symbols different from σ and W is not permuted during the encoding process. Since the top n rows of c_1 and c_2 are identical if they induce B, the remaining argument is analogous to the above proof of Lemma 8. □

A closer look at Lemma 9 shows that we can relax the conditions imposed upon B in that B need not be restricted to a width of two as long as V contains a column vector that is induced at most once in the non-encoding part of B. This finally leads us to a proof of Theorem 5.

Proof. [Theorem 5] Recall the reduction scheme from the proofs of Lemmas 8 and 9. The key point for showing the correctness of the reduction was that, after the reduction, B was not induced in the right part of A. Call this part A_r. According to the prerequisites for this theorem, the submatrix V of height r' has a column vector v that is induced at most once in the non-encoding part of B_σ. Denote the $r \times 2$ submatrix of B_σ that contains both occurences of v in B_σ—the one induced in the encoding and the one induced in the non-encoding part—by $B_{\sigma,v}$.[b] Now, we can simply use the reduction from Lemma 9 and $B_{\sigma,v}$ as the forbidden submatrix to encode an r'-Hitting Set instance into an Row Deletion($B_{\sigma,v}$) instance: Instead of only writing the entries of $B_{\sigma,v}$ into those rows of A determined by the reduction, we write the whole respective rows of B_σ into A. It is easy to verify that this yields a 1:1-correspondence between the solutions of the r'-Hitting Set instance and the Row Deletion(B) instance. □

5. Conclusion

In this work, we have started a systematic study on the complexity of and on algorithms for Row Deletion(B). Among others, we were able to show *NP*-completeness for a number of natural cases of forbidden submatrices B. While our results clearly transfer to the analogous Column Deletion(B) problem, other generalizations are conceivable for which the computational complexity remains to be investigated, such as allowing row- and column deletions at the same time or allowing the modification of entries in the input matrix as in [4].

It also remains open to generalize all special cases treated in this work, for instance, by proving or disproving the following conjecture: For every forbidden submatrix B with at least three rows, Row Deletion(B) is *NP*-complete. One possible lead for attacking this problem might be to extend the hardness results of Yannakakis [23] concerning vertex-deletion problems on bipartite graphs: If the input matrix A and the forbidden submatrix B are binary, A and B can be interpreted as the adjacency matrices of two bipartite graphs. Then, Row Deletion(B) is equivalent to the problem of deleting at most k vertices on one fixed side of A's bipartite graph such that the resulting graph has no subgraph isomorphic to B's

[b]Note that we only need to consider the case where v is induced once in the encoding part of the forbidden submatrix. Otherwise, V cannot be induced in the non-encoding part and B meets the prerequisites of Theorem 4.

graph.

Our work was partially motivated by constructing perfect phylogenies from binary matrices [18, 20]. For this special case, where we have to consider a forbidden submatrix B consisting of the rows $(1,1)$, $(1,0)$, and $(0,1)$, our results yield that Row Deletion(B) is at least as hard as 2-Hitting Set (which is the same as the well-known Vertex Cover problem in graphs) and that it always can be solved by transforming it into an instance of 3-Hitting Set.

Note that there remains an interesting structural "gap" between the tractability and hardness results of this work: Let an $r \times s$ forbidden submatrix B have a σ-decomposition (as defined Definition 3 and illustrated in Figure 2) with height-r' submatrix V. Then, if $r > r'$, we showed, on the one hand, that in certain cases Row Deletion(B) is at least as hard to solve as r'-Hitting Set and, on the other hand, that it is not harder to solve than r-Hitting Set. It would be interesting to determine whether and—if so—how this "hardness localization" can be rendered more precise.

Acknowledgements

The main work on this paper, which originates from the Thesis [22], was done while all authors were with the Wilhelm-Schickard-Institut für Informatik of the Universität Tübingen. Our work was supported by the Deutsche Telekom Stiftung and the Deutsche Forschungsgemeinschaft (DFG) project PEAL (parameterized complexity and exact algorithms, NI 369/1), project OPAL (optimal solutions for hard problems in computational biology, NI 369/2), and Emmy Noether research group PIAF (fixed-parameter algorithms, NI 369/4). We would like to thank an anonymous referee of the IJFCS for careful reading and helpful comments.

References

1. Giorgio Ausiello, Pierluigi Creszenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and Approximation—Combinatorial Optimization Problems and their Approximability Properties.* Springer Verlag, 1999.
2. Andreas Brandstädt, Van B. Le, and Jeremy P. Spinrad. *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications, 1999.
3. Peter Damaschke. On the fixed-parameter enumerability of cluster editing. In *Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science (WG'05)*, volume 3787 of *LNCS*, pages 283–294. Springer Verlag, 2005.
4. Peter Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351(3):337–350, 2006.
5. Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity.* Springer Verlag, 1999.
6. Dorit S. Hochbaum (editor). *Approximation algorithms for NP-hard problems.* PWS Publishing Company, 1997.
7. Henning Fernau. A top-down approach to search-trees: Improved algorithmics for 3-Hitting Set. *Electronic Colloquium on Computational Complexity (ECCC)*, Report

TR04-073, 2004.

8. Eldar Fischer and Ilan Newman. Testing of matrix properties. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC'01)*, pages 286–295, 2001.
9. Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.
10. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
11. Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica*, 39(4):321–347, 2004.
12. Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory of Computing Systems*, 38(4):373–392, 2005.
13. Bettina Klinz, Rüdiger Rudolf, and Gerhard J. Woeginger. Permuting matrices to avoid forbidden submatrices. *Discrete Applied Mathematics*, 60:223–248, 1995.
14. John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.
15. Assaf Natanzon, Ron Shamir, and Roded Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109–128, 2001.
16. Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
17. Rolf Niedermeier and Peter Rossmanith. An efficient fixed-parameter algorithm for 3-Hitting Set. *Journal of Discrete Algorithms*, 1(1):89–102, 2003.
18. Itsik Pe'er, Tal Pupko, Ron Shamir, and Roded Sharan. Incomplete directed perfect phylogeny. *SIAM Journal on Computing*, 33(3):590–607, 2004.
19. Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.
20. Roded Sharan. *Graph Modification Problems and their Applications to Genomic Research*. Ph.D. Thesis, School of Computer Science, Tel-Aviv University, Israel, 2002.
21. Vijay V. Vazirani. *Approximation Algorithms*. Springer Verlag, 2001.
22. Sebastian Wernicke. *On the Algorithmic Tractability of Single Nucleotide Polymorphism (SNP) Analysis and Related Problems*. Diploma Thesis, WSI für Informatik, Universität Tübingen, Germany, 2003.
23. Mihalis Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10(2):310–327, 1981.