

Genetic Algorithms applied to Problems of Forbidden Configurations

R.P. Anstee* Miguel Raggi†

Department of Mathematics
University of British Columbia
Vancouver, B.C. Canada V6T 1Z2

anstee@math.ubc.ca mraggi@gmail.com

Submitted: Jun 27, 2011; Accepted: Nov 15, 2011; Published: Dec 5, 2011

Mathematics Subject Classification: 05C88, 05D05

Abstract

A *simple* matrix is a (0,1)-matrix with no repeated columns. For a (0,1)-matrix F , we say a (0,1)-matrix A *avoids* F (as a *configuration*) if there is no submatrix of A which is a row and column permutation of F . Let $\|A\|$ denote the number of columns of A . We define $\text{forb}(m, F) = \max\{\|A\| : A \text{ is an } m\text{-rowed simple matrix that avoids } F\}$. Define an *extremal matrix* as an m -rowed simple matrix A with that avoids F and $\|A\| = \text{forb}(m, F)$. We describe the use of Local Search Algorithms (in particular a Genetic Algorithm) for finding extremal matrices. We apply this technique to two forbidden configurations in turn, obtaining a guess for the structure of an $m \times \text{forb}(m, F)$ simple matrix avoiding F and then proving the guess is indeed correct. The Genetic Algorithm was also helpful in finding the proof.

Keywords: trace, forbidden configurations, extremal set theory, (0,1)-matrices, Genetic Algorithms

1 Introduction

Define a matrix to be *simple* if it is a (0,1)-matrix with no repeated columns. Then an $m \times n$ simple matrix corresponds to a *simple hypergraph* or *set system* on m vertices with n edges. For a matrix A , let $\|A\|$ denote the number of columns in A . For a (0,1)-matrix F , we define that a (0,1)-matrix A has F as a *configuration*, which we write as $F \prec A$, if there is a submatrix of A which is a row and column permutation of F . Let $\text{Avoid}(m, F)$

*Research supported in part by NSERC

†Research supported in part by NSERC of first author

denote the set of all m -rowed simple matrices A with $F \not\prec A$. Our main extremal problem is to compute

$$\text{forb}(m, F) = \max_A \{\|A\| : A \in \text{Avoid}(m, F)\}.$$

We say a matrix $A \in \text{Avoid}(m, F)$ is an *extremal matrix* if $\|A\| = \text{forb}(m, F)$. We will also consider the natural generalizations where we forbid a family \mathcal{F} of configurations.

In this paper we describe a new computational approach for finding extremal matrices using a Genetic Algorithm as described in Section 2. For a general introduction to Genetic Algorithms one could use [9], [10]. Genetic Algorithms have been applied in a number of Combinatorial problems such as Graph Colouring [6], Travelling Salesman Problem [8] and Steiner Tree Problem [7]. It is quite remarkable that this local search strategy has given such reliable results for us.

We study the following configurations V and W (where $\text{forb}(m, V)$ and $\text{forb}(m, W)$ weren't previously known), and compare to the common submatrix X (where $\text{forb}(m, X)$ was found in [2]).

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \quad (1.1)$$

We use our Genetic Algorithm to seek extremal matrices $A \in \text{Avoid}(m, V)$ and $A \in \text{Avoid}(m, W)$ for small m . From these examples we guess the structure of extremal matrices in general and then we are subsequently able to prove these matrices are indeed extremal. Guessing such structures would have been challenging without the help of a computer. Matrix X above is contained in both V, W and we initially thought (before using the Genetic Algorithm) that $\text{forb}(m, V) = \text{forb}(m, W) = \text{forb}(m, X)$. Related results are in [2],[3]. In particular a list of 3×4 configurations for which the exact bounds are not known are listed in [3].

Theorem 1.1. [2] *Let $m \geq 3$. Then $\text{forb}(m, X) = \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + 1$.*

Theorem 1.2. *Let $m \geq 2$. Then $\text{forb}(m, W) = \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + m - 2$.*

Theorem 1.3. *Let $m \geq 6$. Then $\text{forb}(m, V) = \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + 3$.*

The structure of the apparently extremal matrices A generated by the Genetic Algorithm provided the strategy to tackle the latter two bounds. Interestingly, the Genetic Algorithm was used again in each example in the inductive proof to guess $\text{forb}(m-1, \{\text{a collection of smaller matrices}\})$. We establish Theorem 1.2 in Section 4 and Theorem 1.3 in Section 5. Computer calculations were used during the course of the proof to compute ‘what is missing?’ on each four rows as described in Section 3. All code (C++) used in this paper can be freely downloaded from <http://www.math.ubc.ca/~anstee/>.

Some helpful notation is the following. Let $[m] = \{1, 2, \dots, m\}$. We say a column α has *column sum* ℓ if α has exactly ℓ 1's. Let K_k denote the $k \times 2^k$ simple matrix that consists of all 2^k possible columns in k rows. Let K_k^ℓ denote a $k \times \binom{k}{\ell}$ simple matrix of all

columns of sum ℓ . Let $\mathbf{1}_k$ denote the $k \times 1$ column of 1's and $\mathbf{0}_\ell$ denote the $\ell \times 1$ column of 0's. For two given matrices A, B which have the same number of rows, let $[A | B]$ denote the matrix of A concatenated with B . Let $K_k^{\leq \ell}$ denote the matrix $[K_k^\ell | K_k^{\ell-1} | \cdots | K_k^0]$. For a set of rows S , let $A|_S$ denote the submatrix of A restricted to rows S .

We define $B \times C$ as the *product* of two matrices B, C as follows: Assume B, C have b, c rows respectively. Then $B \times C$ has $b + c$ rows and $\|B\| \cdot \|C\|$ columns. The columns are formed from a column of B on top of a column of C in all possible ways. Column $\mathbf{1}_2 \times \mathbf{0}_2$ would be $(1, 1, 0, 0)^T$. Also $[01] \times K_{k-1} = K_k$. Product constructions were conjectured in [4] to be the asymptotically best constructions avoiding a configuration. A survey on Forbidden Configurations can be found in [1].

2 Genetic Algorithm for Finding Extremal Matrices

In this section, we consider \mathcal{F} a family of (small) configurations and m a (small) fixed integer. Suppose we wish to find (or rather, guess) $\text{forb}(m, \mathcal{F})$ using the computer. The idea is to consider all the 2^m columns and add them one by one into a matrix $A \in \text{Avoid}(m, \mathcal{F})$, making sure at each step we don't create any $F \in \mathcal{F}$ as a configuration. The order in which to add columns is what will determine the size of $\|A\|$ at the end of the process. To do this, enumerate all the columns and for each permutation of $[2^m]$ find the number of columns that would be added while avoiding all $F \in \mathcal{F}$ if we were to add them in that order. So the search space has size $(2^m)!$, which is too big to search exhaustively, but local search can find the correct answer when m isn't too large. We'll describe three methods.

For each permutation, we can separate the columns into *good* columns (which together form a matrix in $\text{Avoid}(m, \mathcal{F})$) and *bad* columns. We wish then to find a permutation that maximizes the number of *good* columns. Perhaps this explanation could benefit by an example. Suppose we wanted to forbid

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and suppose $m = 3$. Consider all eight 3-rowed columns in the following order:

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{array}$$

Then consider the permutation $(4, 1, 3, 6, 5, 8, 2, 7)$. We start adding columns in the order given by this permutation. If adding a certain column would give rise to a copy of F , then don't add it and continue to the next one. Here is how it would work for this particular permutation:

- Add column 4.

$$A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Add column 1:

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

- Can't add column 3.

- Add column 6:

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

- Can't add column 5.

- Add column 8:

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

- Can't add column 2.

- Can't add column 7.

We end up with four *good* columns, $\{4, 1, 6, 8\}$ and four *bad* columns, $\{3, 5, 2, 7\}$.

2.1 Brute Force approach

The brute force method does the following:

- Choose a permutation at random.
- Separate into *good* and *bad* columns.
- Count the number of *good* columns.
- Repeat, while keeping track of the “best” so far (i.e. the one with the most number of *good* columns)

The strength of this method is that we may do this thousands of times in a relatively short time. This method is good for small configurations in which there are many different ways to achieve the bound but, in our experience, there are many cases where this method fails to find an extremal matrix.

2.2 Hill Climbing

This is an improvement over the above described method using the idea of Hill Climbing. Start with some permutation and separate into *good* and *bad* columns, and then for each column b in *bad*, form a new permutation by putting b at the beginning, but leaving all others in place. This ensures that the chosen column b is selected when performing the separation task again for the new permutation. From all possible choices of b , pick the “best one”; that is, the one which gives rise to the most columns in *good*. If there are ties, pick one at random. Repeat this process until there is no improvement for a number of steps. In the example described before, we would have to consider all the following permutations:

1. (3, 4, 1, 6, 5, 8, 2, 7)
2. (5, 4, 1, 3, 6, 8, 2, 7)
3. (2, 4, 1, 3, 6, 5, 8, 7)
4. (7, 4, 1, 3, 6, 5, 8, 2)

In this example all these permutations give four good columns, but in general some might be better than others.

Hill Climbing as described above gets stuck easily, but it does perform better than brute force.

2.3 Genetic Algorithms

An even better way (in practice, we’ve found) to search the space is using a Genetic Algorithm. The idea behind Genetic Algorithms is to mimic evolution. We maintain a population of ‘individuals’ and assign ‘fitness’ to them in some way (in our case, the number of ‘good’ columns). Then pick two at random, but giving higher probability to those with higher fitness. Call them father and mother. Then combine them in some way to produce offspring, hopefully with better results than either father or mother. Do this for many generations.

There are many ways to combine permutations. One that we found that works well for Forbidden Configurations is as follows:

1. For both father and mother, separate the numbers (of the columns) into *good* and *bad*.
2. Choose a random number from the *good* part of mother, and select all numbers (of columns) from the good part starting from the first good column and going up to the chosen number.
3. Permute the entries of father so that the numbers selected from the good part of mother are in the same order as those in mother.

This is better shown with an example. Suppose we had a father and a mother like this:

$$\text{father} = (2, 3, 6, 1 \mid 7, 4, 8, 5) \text{ and mother} = (5, 3, 8, 1, 4 \mid 7, 2, 6)$$

with the numbers shown before the vertical line being the *good* part and the numbers shown after the vertical line being the *bad* part. Pick a random entry in the *good* part of mother. For example, pick 1 and look at the numbers that appear to the left of the picked number. In this case, (5, 3, 8, 1). In father, select these numbers.

$$\text{father} = (2, \mathbf{3}, 6, \mathbf{1} \mid 7, 4, \mathbf{8}, \mathbf{5})$$

Make child by shuffling the selected entries in father so that they match the order of mother.

$$\text{child} = (2, \mathbf{5}, 6, \mathbf{3}, 7, 4, \mathbf{8}, \mathbf{1}).$$

Finally, separate the columns of *child* into *good* and *bad*. This approach has given very good results in our experience. In particular for the two matrices V and W the Genetic Algorithm gave us the matrices which were later proved to be extremal. Then by having the conjectured extremal matrices it was relatively straightforward to construct the proofs. We discuss these proofs in the following sections.

3 Two Proof techniques for Forbidden Configurations

3.1 Standard Induction

The following induction idea has been very successful for studying Forbidden Configurations. Let $A \in \text{Avoid}(m, F)$. Consider deleting a row r . Let $C_r(A)$ be the matrix that consists of the repeated columns of the matrix that is obtained when deleting row r from A . For example, permute the rows of A so that r becomes the first row. Let A' be the matrix obtained from A by deleting row r . Now A' need not be simple and so let $C_r(A)$ denote the largest simple $(m-1)$ -rowed submatrix of A' so that A' contains $[C_r(A)C_r(A)]$. After column permutations, we may decompose A as follows:

$$A = \text{row } r \rightarrow \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 \\ B_r(A) & & C_r(A) & C_r(A) & & D_r(A) \end{bmatrix},$$

where $B_r(A)$ are the columns that appear with a 0 on row r , but don't appear with a 1, and $D_r(A)$ are the columns that appear with a 1 but not a 0 and $C_r(A)$ are the columns that appear with both with a 0 and with a 1 on row r . Thus $[B_r(A)C_r(A)D_r(A)]$ is a simple $(m-1)$ -rowed matrix in $\text{Avoid}(m-1, F)$. We deduce

$$\|A\| \leq \|C_r(A)\| + \text{forb}(m-1, F). \tag{3.1}$$

This means any upper bound on $\|C_r(A)\|$ (as a function of m), automatically yields an upper bound on $\|A\|$ by induction.

Since $C_r(A)$ consists of the repeated columns, we can deduce extra conditions. In particular A contains $[0\ 1] \times C_r(A)$. Thus if for example F' is obtained from F by deleting a row then $C_r(A) \in \text{Avoid}(m-1, F')$. Also if $F \prec [F''|F''']$, then $C_r \in \text{Avoid}(m-1, F''')$. We search for a row r of A such that $\|C_r(A)\|$ is as small as possible. If we could prove that there always exists a row r such that $\|C_r(A)\|$ is small enough, we could proceed by induction.

3.2 What is Missing?

Suppose F , the forbidden configuration, has s rows and suppose A doesn't contain F . Consider all s -tuples of rows from A and for each s -tuple of rows S , consider the submatrix $A|_S$ of A formed from rows S of A . For example, if $s = 2$ and $S = \{2, 4\}$ and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

then

$$A|_S = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Without any restriction, A_S could have all $2^{|S|}$ possible columns, each appearing multiple times. But with the restriction that $F \not\prec A$, in particular $F \not\prec A|_S$, so some columns have to be missing; we can't have all 2^s columns appear in $A|_S$ as many times as they appear in F . Some may not appear at all. In our example, the column $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ appears only once in $A|_S$, while $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ both appear 3 times, but $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ doesn't appear at all.

We wrote a program whose input is a configuration F (or even a family of configurations \mathcal{F}), and its output is the list of possibilities for columns absent or appearing a restricted number of times. Studying this list often is often easier than studying F for the purpose of analyzing what the structure of a matrix in $\text{Avoid}(m, F)$ is. Unfortunately, the program performs doubly exponential ($O(2^{2^s})$) matrix comparison operations, where s is the number of rows on which we wish to find what is missing. Furthermore, each matrix comparison takes $O(s!)$ time. In practice, this means checking ≤ 4 rows is instantaneous, 5 rows takes, depending on the configuration, anywhere from a few minutes to a couple of hours, and with 6 rows we would have to make $\sim 2^{64}$ configuration comparison operations, which is many orders of magnitude more than any computer could handle.

4 Exact bound for W

We'll prove that $\text{forb}(m, W) = \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + m - 2$. We can check that the first few cases (for $m \leq 5$) work using a computer. Then we proceed by the Standard Induction: Let $A \in \text{Avoid}(m, F)$. If we could find some row r for which $\|C_r(A)\| \leq m + 1$, then we could conclude using 3.1 that

$$\begin{aligned} \|A\| &\leq m + 1 + \binom{m-1}{2} + \binom{m-1}{1} + \binom{m-1}{0} + (m-1) - 2 \\ &= \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + m - 2. \end{aligned}$$

So we may assume that $\|C_r(A)\| \geq m + 2$ for every r .

Since $C_r(A)$ consists of repeated columns of A , note it cannot contain any of the following three configurations:

$$G_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad G_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Given that $C_r(A) \in \text{Avoid}(m-1, \{G_1, G_2, G_3\})$, the following Lemma establishes that $\|C_r(A)\| \leq (m-1) + 3 = m + 2$.

Lemma 4.1. *Let $m \geq 4$. Then $\text{forb}(m, \{G_1, G_2, G_3\}) = m + 3$.*

Proof: For the lower bound $\text{forb}(m, \{G_1, G_2, G_3\}) \geq m + 3$ an example (which was found again using our Local Search strategies) suffices. Consider the matrix

$$A = [\mathbf{0}_m \mid K_m^1 \mid \mathbf{1}_m \mid \alpha],$$

where α is any other column. Clearly $A \in \text{Avoid}(m, \{G_1, G_2, G_3\})$ and $\|A\| = m + 3$. To prove the upper bound, we proceed by induction on m . Let $A \in \text{Avoid}(m, \{G_1, G_2, G_3\})$. Then, if we forbid $\{G_1, G_2, G_3\}$, below are the 16 possible cases of what is missing on each quadruple of rows.

$$\begin{array}{cccccccccccccccc} \text{no} & \leq 1 & > 1 & > 1 \\ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

$$\begin{array}{cccccccccccccccc} \text{no} & \leq 1 & > 1 & > 1 \\ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

$$\begin{array}{cccccccccccccccc} \text{no} & \leq 1 & > 1 & > 1 \\ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array}$$

no	≤ 1	≤ 1	≤ 1	> 1	> 1										
$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

no	≤ 1	≤ 1	≤ 1	> 1	> 1										
$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

no	≤ 1	≤ 1	> 1	> 1											
$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

no	≤ 1	≤ 1	> 1	> 1											
$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

no	≤ 1	≤ 1	> 1	> 1											
$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

We easily check this list to see that there are at most seven columns present on four rows (at least 9 are absent) and so $\text{forb}(4, \{G_1, G_2, G_3\}) = 7$. Now consider $m \geq 5$. It's easy to check that in every quadruple there is a row (the final row will always work) and a column we can delete from A and keep the remaining $(m - 1)$ -rowed matrix A' simple. Then by induction, $\|A'\| \leq (m - 1) + 3 = m + 2$ and then $\|A\| \leq \|A'\| + 1 \leq m + 3$. To find such a row and column, look at the columns marked ≤ 1 and > 1 , and see that there is a row we can delete such that the only repeat (if there is one) has one of the columns marked ≤ 1 . We used the help of the free software sage (<http://www.sagemath.org/>), but it could also be done by hand. For example, for the first possibility,

$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{ccccccc}
 \leq 1 & > 1 & > 1 \\
 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}
 \end{array}
 ,$$

deleting from A the row ℓ and the column of A corresponding to the fifth column above is enough to keep the resulting matrix A' simple. Thus $\|A\| \leq m + 3$. ■

We need a more detailed lemma about the structure of any $m \times (m + 3)$ matrix in $\text{Avoid}(m, \{G_1, G_2, G_3\})$. This was predicted using our Genetic Algorithm too.

Lemma 4.2. *Let $A \in \text{Avoid}(m, \{G_1, G_2, G_3\})$ with $m \geq 5$ and $\|A\| = m + 3$. Then $K_m^1 \prec A$. Moreover, the remaining three columns are $\mathbf{0}_m$ and two additional columns α, β with $\alpha < \beta$ (meaning that on each row for which α has a 1, β also has a 1).*

Proof: We proceed by induction on m . We checked all cases with $m \leq 5$ using a computer. Assume $A \in \text{Avoid}(m, \{G_1, G_2, G_3\})$ with $\|A\| = m + 3$ and $m \geq 6$. The fact that A has no G_1 means the columns, when considered as sets, form a *laminar family*: Namely for any two columns α, β , either $\alpha < \beta$ or $\beta < \alpha$ or α, β have no row in which both have 1's. Sometimes the term noncrossing is used. Viewing the columns as sets, we are asking that for any two sets A, B , either $A \subset B$, $B \subset A$ or $A \cap B = \emptyset$.

From our proof of Lemma 4.1, there is a row and a column from A we can delete to obtain a $(m - 1) \times (m + 2)$ simple matrix A' . We may assume $K_{m-1}^1 \prec A'$ by induction. Assume we deleted the last row from A to obtain A' and that the deleted column was the last column of A . If we restrict ourselves to the first $m - 1$ rows, the deleted column has to be a repeat of one of the columns of A' , else we would have an $(m - 1) \times (m + 3)$ simple matrix contradicting Lemma 4.1.

At this point the proof is finished save for a case analysis on each of the possible columns to repeat. In each case we repeat a column and attempt to find either G_1 , G_2 or G_3 . This case analysis is probably much easier as an exercise for the reader than it is to either write or read about. Given that no interesting ideas appear in this admittedly tedious but very straightforward case analysis, we have moved it to Appendix A for the interested reader.

We are now ready to prove Theorem 1.2.

Proof of Theorem 1.2: We use induction on m . The result is true for $m = 3$ so we may assume $m \geq 4$. Let $A \in \text{Avoid}(m, W)$. Apply the decomposition (3.1). If $\|C_r(A)\| \leq m + 1$, then we apply (3.1) to establish the bound for $\|A\|$ by induction. So assume $\|C_r(A)\| \geq m + 2$ for all choices of r . By Lemma 4.1, we deduce that $\|C_r(A)\| = m + 2$ for every row r , and by Lemma 4.2, we have that $K_{m-1}^1 \prec C_r(A)$ also for every row r . Thus $K_m^2 \prec A$, since all columns of column sum 1 in $C_r(A)$ appear with a 1 in row r (and this happens for every row r). We also have $[K_m^1 \mid K_m^0] \prec A$. Now in every triple of rows of K_m^2 we have the matrix G_1 once in every ordering of the triple. Given that $W = [G_1 \mid G_1]$, the columns of A of column sum at least 3 have no configuration G_1 . We appeal to Lemma 4.3 below to deduce that the number of columns in A of column sum at least 3 is at most $m - 2$. Then $\|A\| \leq \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + m - 2$ which yields the desired bound. ■

A quick counting argument reveals the following general result about laminar families which we use in the proof of Theorem 1.2.

Lemma 4.3. *Let $m \geq 3$ and let \mathcal{Z} be a laminar family of subsets of $[m] = \{1, 2, \dots, m\}$ with the property that for all $Z \in \mathcal{Z}$ we have $|Z| \geq 3$. Then $|\mathcal{Z}| \leq m - 2$ and furthermore,*

this bound is tight (i.e. there exist a family \mathcal{Z} for which $|\mathcal{Z}| = m - 2$). Thus, if $A \in \text{Avoid}(m, F)$ satisfies that all column sums are at least 3, then $\|A\| \leq m - 2$.

Proof: Let $f(x)$ denote the size of the largest laminar family of $[x]$ with no sets of size less than or equal to 2. Assume $A \in \text{Avoid}(m, F)$ with the property that all column sums are at least 3. We wish to show that $f(m) = m - 2$.

The family $\{[3], [4], \dots, [m]\}$ has size $m - 2$, which proves $f(m) \geq m - 2$. We wish to prove $f(m) \leq m - 2$.

Let \mathcal{Z} be such that $|\mathcal{Z}| = f(m)$. We may assume $[m] \in \mathcal{Z}$: if $[m] \notin \mathcal{Z}$, observe that $\mathcal{Z} \cup \{[m]\}$ is also a laminar family of bigger size. Suppose then that the next biggest set Z in \mathcal{Z} has size k . We partition $[m]$ into two disjoint sets: Z and $[m] \setminus Z$. Every set $Y \in \mathcal{Z}$ satisfies either $Y \subseteq Z$ or $Y \subseteq [m] \setminus Z$ or $Y = [m]$. This gives the recurrence

$$f(m) \leq 1 + f(k) + f(m - k).$$

If $k \neq m - 1$, then by induction $f(k) = k - 2$ and $f(m - k) = m - k - 2$, and so $f(m) \leq 1 + k - 2 + m - k - 2 = m - 3$, a contradiction. When $k = m - 1$ we have $f(m) \leq m - 2$. Moreover if $f(m) = m - 2$ we observe that \mathcal{Z} is “equivalent” to $\{[3], [4], \dots, [m]\}$. ■

5 Exact bound for V

Using the computer, we can prove by exhaustively looking at all the possibilities that $\text{forb}(3, V) = 8$, $\text{forb}(4, V) = 13$, $\text{forb}(5, V) = 18$. Using the Genetic Algorithm of Section 2, we obtained large matrices with no subconfiguration V which suggested to us that $\text{forb}(6, V) = 25$, $\text{forb}(7, V) = 32$, $\text{forb}(8, V) = 40$. For $m \geq 6$, this suggests $\text{forb}(m, V) = \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + 3$, two more than our first guess that $\text{forb}(m, V) = \text{forb}(m, X)$.

The Genetic Algorithm also predicted the structure of matrices in $\text{ext}(m, V)$. Following from the matrices discovered by the Genetic Algorithm, we will consider the following matrices in $\text{Avoid}(m, V)$: Consider a partition of the m rows into two disjoint sets, say $U = \{1, 2, \dots, u\}$ and $L = \{u + 1, u + 2, \dots, m\}$. Suppose $|U| = u$ and $|L| = \ell$ with $u + \ell = m$. Let A' have the following structure:

$$A' = \begin{array}{c} U \\ L \end{array} \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} \mathbf{0}_u & \mathbf{0}_u & \mathbf{0}_u & \mathbf{0}_u & K_u^1 & \mathbf{1}_u & K_u^1 & K_u^2 & \mathbf{1}_u \\ \times & \times \\ \mathbf{0}_\ell & \mathbf{1}_\ell & K_\ell^{\ell-1} & K_\ell^{\ell-2} & K_\ell^{\ell-1} & \mathbf{0}_\ell & \mathbf{1}_\ell & \mathbf{1}_\ell & \mathbf{1}_\ell \end{array} \right]. \quad (5.1)$$

We easily check that $A' \in \text{Avoid}(m, V)$ and for $3 \leq u, \ell \leq m - 3$ we have $\|A'\| = \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + 3$. We will prove that $A' \in \text{ext}(m, V)$ and hence establish Theorem 1.3. To prove this, consider some $A \in \text{Avoid}(m, V)$ and apply the standard decomposition 3.1. We deduce that $C_r(A)$ can't have any of the three following configurations:

$$H_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad H_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

no	≤ 1	≤ 1	≤ 1	≤ 1	> 1											
$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$
no	≤ 1	≤ 1	≤ 1	≤ 1	> 1											
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$
no	≤ 1	≤ 1	> 1	> 1												
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$
no	≤ 1	≤ 1	≤ 1	> 1												
$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
no	≤ 1	≤ 1	≤ 1	> 1												
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

This verifies in particular that $\text{forb}(4, \{H_1, H_2, H_3\}) = 6$ (at least 10 columns out of 16 are absent in each of the twenty cases). We proceed as we did in Lemma 4.1 and verify that each of these possibilities yields one row and at most one column that we can delete from A , maintaining simplicity. This means that $\|A\| \leq m + 2$ by induction. ■

Lemma 5.2. *Assume $m \geq 4$. Let $A \in \text{Avoid}(m, \{H_1, H_2, H_3\})$ with $\|A\| = m + 2$. Then there exists a partition $U \subseteq [m]$ and $L = [m] \setminus U$ with $|U| = u \geq 1$ and $|L| = \ell = m - u \geq 1$ (U for Upper and L for Lower) so that if we permute rows,*

$$A = \begin{array}{c} U \\ L \end{array} \left[\begin{array}{c|c|c|c} \mathbf{0}_u & K_u^1 & \mathbf{0}_u & \mathbf{0}_u \\ \times & \times & \times & \times \\ \mathbf{1}_\ell & \mathbf{1}_\ell & K_\ell^{\ell-1} & \mathbf{0}_\ell \end{array} \right] \quad \text{or} \quad A = \begin{array}{c} U \\ L \end{array} \left[\begin{array}{c|c|c|c} \mathbf{0}_u & K_u^1 & \mathbf{0}_u & \mathbf{1}_u \\ \times & \times & \times & \times \\ \mathbf{1}_\ell & \mathbf{1}_\ell & K_\ell^{\ell-1} & \mathbf{1}_\ell \end{array} \right] \quad (5.2)$$

Note that in the former case we must have $\ell \geq 2$ and in the latter case $u \geq 2$.

Proof: We use induction on m . The result is true for $m = 4$. Assume $m \geq 5$. Let $A \in \text{Avoid}(m, \{H_1, H_2, H_3\})$. By our argument in Lemma 5.1, there is a row and a single

column we can delete, leaving the remainder simple. Let A' be the resulting simple matrix. We may assume by induction that there exists disjoint sets U', L' such that $|U'| = a \geq 1$, $|L'| = b \geq 1$ where $a + b = m - 1$ so that after permuting rows and columns,

$$A' = \begin{array}{c} U' \\ L' \end{array} \left[\begin{array}{c|c|c|c} \mathbf{0}_a & K_a^1 & \mathbf{0}_a & \mathbf{0}_a \\ \times & \times & \times & \times \\ \mathbf{1}_b & \mathbf{1}_b & K_b^{b-1} & \mathbf{0}_b \end{array} \right], \text{ or } A' = \begin{array}{c} U' \\ L' \end{array} \left[\begin{array}{c|c|c|c} \mathbf{0}_a & K_a^1 & \mathbf{0}_a & \mathbf{1}_a \\ \times & \times & \times & \times \\ \mathbf{1}_b & \mathbf{1}_b & K_b^{b-1} & \mathbf{1}_b \end{array} \right].$$

Without loss of generality, we may assume $b \geq 2$. Assume the last column ($m + 2$) and last row (m) of A is deleted to obtain A' . After deleting row m , the last column of A must be one of the columns of A' given that $\|A'\| = \text{forb}(m, \{H_1, H_2, H_3\})$. In order to avoid H_1, H_2 and H_3 in A , we can show that we have the desired structure (5.1) in A with either a or b one larger.

As in the proof of Theorem 1.2, we must do some tedious but straightforward case analysis for which column gets repeated. We move the gory details of the case analysis to Appendix B. Again, this case analysis is much easier to do by oneself than it is to either write or read about. ■

Proof of Theorem 1.3: We use induction on m for $m \geq 6$. We established by computer that $\text{forb}(5, V) = 18$ (which is smaller than the bound of Theorem 1.3). Noting that $\text{forb}(5, \{H_1, H_2, H_3\}) = 7$, we deduce using (3.1) that

$$\text{forb}(6, V) \leq \text{forb}(5, V) + \text{forb}(5, \{H_1, H_2, H_3\}) = 18 + 7 = 25$$

and so $\text{forb}(6, V) = 25$, because of construction (5.1). Thus, we may assume $m \geq 7$.

By induction, assume $\text{forb}(m-1, V) = \binom{m-1}{2} + \binom{m-1}{1} + \binom{m-1}{0} + 3$. Let $A \in \text{Avoid}(m, V)$ with $\|A\| = \text{forb}(m, V)$. Apply the standard decomposition (3.1) to A for some row r . If $\|C_r(A)\| \leq m$, we obtain

$$\begin{aligned} \|A\| &\leq \|C_r(A)\| + \text{forb}(m-1, V) \\ &\leq m + \binom{m-1}{2} + \binom{m-1}{1} + \binom{m-1}{0} + 3 \\ &= \binom{m}{2} + \binom{m}{1} + \binom{m}{0} + 3 \end{aligned}$$

Thus, we may assume that for every r we have $\|C_r(A)\| \geq m + 1$. Using Lemma 5.1 (with m replaced by $m - 1$), we may assume $\|C_r(A)\| = m + 1$ for each r . Then using Lemma 5.2 we can assume $C_r(A)$ has the structure of (5.2) for every r so that every $C_r(A)$ partitions $[m] \setminus r$ rows into sets U_r, L_r with $|U_r|, |L_r| \geq 2$. Note also that the only difference between the two possible structures is a column of 0's or a column of 1's neither of which is used in the case analysis below. Furthermore, we will prove that there is a partition of the rows $[m]$ of A into U, L where $U_r = U \setminus r$ and $L_r = L \setminus r$.

Take two rows, say s and t . Consider $C_s(A)$ and $C_t(A)$ as determined by (3.1) by setting $r = s$ and $r = t$. Applying Lemma 5.2 when considering $C_s(A)$ and $C_t(A)$ we

obtain the partitions U_s, L_s, U_t, L_t of rows. Remember that $C_s(A)$ and $C_t(A)$ both appear twice in A with 0's and 1's in rows s and t respectively. We now define partitions $U'_s = U_s \setminus t, L'_s \setminus t, U'_t \setminus s, L'_t \setminus s$ so that $U'_s \cup L'_s = [m] \setminus \{s, t\} = U'_t \cup L'_t$. We assumed $m \geq 7$ and so $|[m] \setminus \{s, t\}| \geq 5$. Hence we may assume that at least one of U'_s and L'_s has size at least 3. Without loss of generality, assume $|U'_s| \geq 3$.

Let

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

Consider the following three cases:

$|U'_s \cap L'_t| \geq 3$: We can find V in rows $U'_s \cap L'_t$ in A (since A contains two copies of K_3^1 in each triple of rows of U'_s and two copies of K_3^2 in each triple of rows of L'_t).

$|U'_s \cap L'_t| = 2$: Then $|U'_s \cap U'_t| \geq 1$, and so we can find V in $A|_{U'_s}$ by taking two rows of $U'_s \cap L'_t$ together with any row in the intersection $U'_s \cap U'_t$. We find Y as a submatrix in any row order (A contains two copies of K_3^1 in each triple of rows of U'_s) and we also have X as a submatrix whose first two rows are from $U'_s \cap L'_t$ and the last one from $U'_s \cap U'_t$. This yields V .

$|U'_s \cap L'_t| = 1$: We have $|U'_s \cap U'_t| \geq 2$, and so we can find V in $A|_{U'_s}$ by taking the row of $U'_s \cap L'_t$ together with two rows in the intersection $U'_s \cap U'_t$. We find Y as a submatrix in any row order (A contains two copies of K_3^1 in each triple of rows of U'_s) and we also have X as a submatrix whose first row is $U'_s \cap L'_t$ and last two rows are from $U'_s \cap U'_t$. This yields V .

This means that $|U'_s \cap L'_t| = 0$ and so $U'_s \subset U'_t$, but then $|U'_t| \geq 3$ and so analogously $U'_t \subset U'_s$. So $U'_s = U'_t$, and then $L'_s = L'_t$. The same conclusion will hold if $|L'_s| \geq 3$. Thus for all $s, t \in [m]$, $U'_s = U'_t$, and then $L'_s = L'_t$.

Using $m \geq 4$, we may now deduce that there is a partition U, L of $[m]$ so that for any r , $U_r = U \setminus r$ and $L_r = L \setminus r$. This proves that the partition for each C_r is really a global partition. Let $|U| = u$ and $|L| = \ell$. We know $u, \ell \geq 2$ since for example if $|U'_r| \geq 1$ and $U'_r = \{s\}$ then $U'_s \cup \{s\} \subseteq U$ and we have $|U| \geq 2$.

Note that for every row r , we have that $[0 \ 1] \times C_r(A) \prec A$. We deduce A contains the following columns:

$$B = \begin{array}{c} U \\ L \end{array} \left[\begin{array}{c|c|c|c|c|c|c|c} \mathbf{0}_u & \mathbf{0}_u & \mathbf{0}_u & \mathbf{0}_u & K_u^1 & K_u^1 & K_u^2 & \mathbf{1}_u \\ \times & \times \\ \mathbf{0}_\ell & \mathbf{1}_\ell & K_\ell^{\ell-1} & K_\ell^{\ell-2} & K_\ell^{\ell-1} & \mathbf{1}_\ell & \mathbf{1}_\ell & \mathbf{1}_\ell \end{array} \right]. \quad (5.3)$$

We have included the column of 0's and the column of 1's since such columns can be added to any matrix without creating V . What other columns might we add to this? For $u \geq 3$, matrix B contains

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

in any triple of rows or U in any row order. So $(A-B)|_U$ must not contain the configuration $(1, 1, 0)^T$, else A has subconfiguration V . Similarly for $\ell \geq 3$, $(A-B)|_L$ must not contain the configuration $(1, 0, 0)^T$.

Thus, for $u, \ell \geq 3$, all columns of $(A-B)$ are in $[\mathbf{0}_u \mid K_u^1 \mid \mathbf{1}_u] \times [\mathbf{0}_\ell \mid K_\ell^{\ell-1} \mid \mathbf{1}_\ell]$. The only column not already in B is $\mathbf{1}_u \times \mathbf{0}_\ell$ which is a column of the hypothesized structure (5.1). Thus, without loss of generality, we need only consider the case $u = 2, \ell \geq 5$. Let $U = \{a, b\}$ and consider any two rows $c, d \in L$. We know B contains $K_u^1 \times [\mathbf{1}_\ell \mid K_\ell^{\ell-1}]$ and $[\mathbf{0}_u \mid K_u^1] \times K_\ell^{\ell-1}$. So B has:

$$\begin{array}{c|cccccc} a & 0 & 0 & 1 & 0 & 0 & 1 \\ b & 1 & 1 & 0 & & & \\ \hline c & 1 & 1 & 0 & 0 & 0 & 1 \\ d & & & & 1 & 1 & 0 \end{array} .$$

Note we can interchange a with b and c with d . To avoid V we must not have columns in $(A-B)$ with

$$\begin{array}{c|cc} a & 1 & 1 \\ b & 0 & \\ \hline c & 0 & 1 \\ d & & 0 \end{array} .$$

Thus, the only possible columns of $A-B$ are

$$\begin{array}{c|ccccc} a & 0 & 1 & 0 & 1 & 1 \\ b & 0 & 0 & 1 & 1 & 1 \\ \hline L & \alpha & \mathbf{1}_\ell & \mathbf{1}_\ell & \mathbf{0}_\ell & \mathbf{1}_\ell \end{array} .$$

where α is any column. Recall that since $\ell \geq 3$, any such α must avoid configuration $(1, 0, 0)^T$. All these columns are already in B , except for $\mathbf{1}_2 \times \mathbf{0}_\ell$, which together with B completes the hypothesized structure (5.1). The desired bound follows. ■

Interestingly, the structure of (5.1) falls short of the bound in the case $u = 2$.

References

- [1] R.P. Anstee, A Survey of Forbidden Configurations results, <http://www.math.ubc.ca/~anstee>.
- [2] R.P. Anstee, F. Barekat and A. Sali, Small Forbidden Configurations V: Exact Bounds for 4×2 cases, *Studia Sci. Math. Hun.*, 48(2011), 1-22.
- [3] R.P. Anstee, S.N. Karp, Forbidden Configurations: Exact bounds determined by critical substructures, *Elec. J. of Combinatorics*, 17 (2010), R50, 27pp.
- [4] R.P. Anstee and A. Sali, Small Forbidden Configurations IV, *Combinatorica*, 25(2005), 503-518.
- [5] Barricelli, Nils Aall, Symbiogenetic evolution processes realized by artificial methods. *Methodos*, 9(1957) 143-182.

- [6] Charles Fleurent and Jacques Ferland, *Genetic and hybrid algorithms for graph coloring*, Annals of Operations Research **63** (1996), 437–461.
- [7] A. Kapsalis, V. J. Rayward-Smith, and G. D. Smith, *Solving the graphical Steiner tree problem using Genetic Algorithms*, The Journal of the Operational Research Society **44** (1993), no. 4, pp. 397–406.
- [8] P. Larrañaga, C. M. H. Kuijpers, R.H. Murga, I. Inza, and S. Dizdarevic, *Genetic algorithms for the travelling salesman problem: A review of representations and operators*, Artificial Intelligence Review **13** (1999), 129–170.
- [9] Melanie Mitchell, *An introduction to Genetic Algorithms*, MIT Press, Cambridge, MA., 1996.
- [10] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee, *A field guide to genetic programming*, Lulu Enterprises, UK Ltd, 2008.

A Case analysis for W

Recall that in order to prove Lemma 4.2, only a case analysis for which column was “repeated” is needed. Let $A \in \text{Avoid}(m, \{G_1, G_2, G_3\})$. We deduced that there is a row and a column of A that we can delete and maintain simplicity. Without loss of generality, suppose it is the last row and the last column of A . Let A' be A without the last row and the last column, which we are assuming by the induction hypothesis to have the desired structure (described in Lemma 4.2). Recall that aside from the column of zeros and the columns of K_{m-1}^1 , there are two columns α, β in A' with $\alpha < \beta$. We call α the *small* column and β the *big* column. We consider 3 different types of rows of A' :

- Row type 1: Both α, β has 0 in the row.
- Row type 2: Column β has a 1 and α has a 0 in the row.
- Row type 3: Both α, β has 1 in the row.

There may not be any rows of type 1, but there has to be at least one row of type 2 and at least two rows of type 3. Consider the generic rows below. We’ve included the appropriate parts of the copy of K_{m-1}^1 and the column of 0’s. The entries marked c_1, c_2, \dots, r_4 are the entries of the deleted row and column.

							α	β		
0	1	0	0	0	0	0	r_1		type 1	
0	0	1	0	0	0	1	r_2		type 2	
0	0	0	1	0	1	1	r_3		type 3	
0	0	0	0	1	1	1	r_4		type 3	
c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8			

Of course there might be many rows of each of the types 1,2,3, but there is no loss of generality if we focus on these rows. There have to be at least two rows of type 3 so it is

possible to have two rows which correspond to the entries r_3, r_4 . We have to be careful because row r_1 might not exist.

Case 1: 0_{m-1} is the repeated column. Then we have

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & 1 \end{array} .$$

So either $K_m^1 \prec A$ or at least one of c_2, c_3, c_4, c_5 is 1. If $c_2 = 1$ then $c_6 = 0$ and $c_7 = 0$ in order to have a laminar family. But then we have G_2 in the last and next-to-last rows. So we may assume $c_2 = 0$. If $c_3 = 1$ then $c_5 = 0, c_6 = 0$ and $c_7 = 1$ by the laminar property and we have then

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & c_4 & 0 & 0 & 1 & 1 \end{array} .$$

But then we have G_2 in the last two rows. We may then assume that $c_3 = 0$. If $c_4 = 1$ then both c_6 and c_7 have to be 1, and so we get

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & c_5 & 1 & 1 & 1 \end{array} ,$$

which has G_3 in the last two rows. This completes Case 1.

Case 2: The repeated column has column sum 1. There are three sub-cases, depending on the position of the 1 in the new column. Let r be the row on which, other than the last row, the new column has a 1.

Subcase 2a: r is of type 1. We have

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline c_1 & 0 & c_3 & c_4 & c_5 & c_6 & c_7 & 1 \end{array} ,$$

which contains G_2 in the first two rows.

Subcase 2b: r is of type 2. We have

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline c_1 & c_2 & 0 & c_4 & c_5 & c_6 & c_7 & 1 \end{array},$$

which contains G_2 in the second and third rows.

Subcase 2c: r is of type 3. We have

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline c_1 & 0 & c_3 & c_4 & c_5 & c_6 & c_7 & 1 \end{array},$$

which contains G_3 in the third and fourth row.

Case 3: The repeated column is the *small* column α . Then we have this:

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline c_1 & c_2 & c_3 & c_4 & c_5 & 0 & c_7 & 1 \end{array}.$$

So c_7 has to be 1 in order to have a laminar family. If either c_4 or c_5 were 0, then we would have G_3 in the last row together with one of the next-to-last rows. So both have to be 1. But this contradicts that we have a laminar family.

Case 4: The repeated column is the *big* column β : Then we have this:

$$\begin{array}{ccccccc|c} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & 0 & 1 \end{array}.$$

This yields G_3 in the second and third rows. This completes all cases.

Now if $K_m^1 \prec A$, we readily deduce that A has $\mathbf{0}_m$ and for any pair of rows i, j , we have K_2^1 in rows i, j of K_m^1 . Hence to avoid G_2 , there is no configuration K_2^1 in the remaining columns α, β and so we deduce that either $\alpha < \beta$ or $\beta < \alpha$. ■

B Case analysis for V

Recall that we had a matrix $A \in \text{Avoid}(m, \{H_1, H_2, H_3\})$ and we deduced there was a row and a column of A we could delete and keep simplicity. Without loss of generality, suppose the deletable row and column are the last row and last column of A . We let A' be A without the last row and the last column. By the induction hypothesis, we assume A' has the desired structure described in Lemma 5.2.

We can make a few general comments about row m , the deletable row of A . If we have both a 0 and a 1 in row m under the columns containing K_a^1 , then using the two columns containing the 0 and 1 and two rows of the U' (we may assume $a \geq 2$ in this case because of the two columns) together with row m , we obtain a copy of H_1 , a contradiction. Similarly we cannot have both a 0 and a 1 in row m under the columns containing K_b^{b-1} .

It is also true that in row m we cannot have a 0 under K_a^1 and a 1 under K_b^{b-1} else we find a copy of H_1 in two columns containing 0 and 1 in row m and in a row of U' , a row of L' and row m .

We will consider the two cases that A' has either the column $\mathbf{0}_{m-1}$ or $\mathbf{1}_{m-1}$ together. Note that for $a = 1$, then A' has both $\mathbf{0}_{m-1}$ and $\mathbf{1}_{m-1}$.

Let γ be the last column of A , which we are assuming that, except for row m , γ is exactly equal to some other column of A' . We analyze all possibilities for γ .

Case 1: $\gamma = \mathbf{0}_a \times \mathbf{1}_b$. We deduce that column $\mathbf{0}_{m-1}$ of A' (if present) appears with a 0 in row m else we have a copy of H_1 in two rows of L' and row m . Similarly, if column $\mathbf{1}_{m-1}$ is present then $a \geq 2$ and appears with a 1 in row m else we have a copy of H_1 in two rows of U' and row m .

Now if we have 1's in row m under both K_a^1 and K_b^{b-1} , then we have the desired structure with $U = U'$ and $L = L' \cup \{m\}$. Similarly, if we have 0's in row m under both K_a^1 and under K_b^{b-1} then we have the desired structure with $U = U' \cup \{m\}$ and $L = L'$.

Thus we may assume we have 1's under K_a^1 and 0's under K_b^{b-1} . We have two subcases.

Subcase 1a: A' has $\mathbf{0}_{m-1}$ (and so A has $\mathbf{0}_m$). We find a copy of H_3 in A in a row of L' and row m in the column $\mathbf{0}_m$ of A , a column from $\mathbf{0}_a \times K_b^{b-1} \times \mathbf{0}_1$ with a 0 in the chosen row of L' , the column with $\mathbf{0}_a \times \mathbf{1}_b \times \mathbf{0}_1$ and a column from $K_a^1 \times \mathbf{1}_{b+1}$.

Subcase 1b: A' has $\mathbf{1}_{m-1}$ (and so A has $\mathbf{1}_m$). If $a = 1$ we have that A' has $\mathbf{0}_{m-1}$ and so we use Subcase 1a. Assuming $a \geq 2$ we find a copy of H_3 in A in a row of U' and row m in the column $\mathbf{1}_m$ of A , a column from $K_a^1 \times \mathbf{1}_{b+1}$ with a 1 in the chosen row of U' , the column with $\mathbf{0}_a \times \mathbf{1}_b \times \mathbf{0}_1$ and a column from $\mathbf{0}_a \times K_b^{b-1} \times \mathbf{0}_1$.

This completes Case 1.

Case 2: $\gamma = \mathbf{0}_a \times \beta_b$, where $\beta_b \in K_b^{b-1}$. Given that $\mathbf{0}_a \times \beta_b$ is repeated in A on rows $[m-1]$, we deduce that it appears both with a 1 and with a 0 on row m . Assume any other column $\mathbf{0}_a \times \beta'_b$ of A' (with $\beta'_b \in K_b^{b-1}$) has a 0 in row m . Then we find a copy of H_1 in row m and the two rows of L' where β_b and β'_b differ and in the column of A containing $\mathbf{0}_a \times \beta'_b$ and one of the two columns of A containing $\mathbf{0}_a \times \beta_b$ which differs from the column containing $\mathbf{0}_a \times \beta'_b$ in row m . If all entries of row m under K_b^{b-1} are 1, then we find H_1 by choosing column γ and any other column in $\mathbf{0}_a \times K_b^{b-1}$, together with the corresponding rows in L' (the one where γ has a 0 and the one where the other column has a 0), together with row m .

Case 3: $\gamma = \alpha_a \times \mathbf{1}_b$, where $\alpha_a \in K_a^1$. For $a \geq 2$ we may follow the argument of the previous case and find a copy of H_1 . Given $a = 1$, we have that A' contains $\mathbf{0}_{m-1}$ as well as $\alpha_a \times \mathbf{1}_b = \alpha_1 \times \mathbf{1}_{m-2} = \mathbf{1}_{m-1}$. We deduce that A has $\mathbf{1}_m$ and $\mathbf{1}_{m-1} \times \mathbf{0}_1$. Thus we can find a copy of H_3 in a row of U' together with a row of L' , in the two columns with $\mathbf{1}_{m-1}$ in A' , the column with $\mathbf{0}_{m-1}$ in A' and a column of $\mathbf{0}_1 \times K_b^{b-1}$ selected in order to have a 0 on the chosen row of L' .

Case 4: $\gamma = \mathbf{0}_a \times \mathbf{0}_b$. We find H_3 in two rows of L' if $b \geq 2$.

Case 5: $\gamma = \mathbf{1}_a \times \mathbf{1}_b$. If $a \geq 2$, we can find H_3 in two rows of U' . In case $a = 1$, we know $\mathbf{0}_{m-1}$ as well as $\mathbf{1}_{m-1}$ are in A' . We find a copy of H_3 using a row of U' and a row of L' where we choose a column of $\mathbf{0}_a \times K_b^{b-1}$ that has a 0 in the chosen row of L' .

This completes the proof of Lemma 5.2. ■