Binary Matrix Factorisations Under Boolean Arithmetic

Réka Ágnes Kovács

Keble College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary Term 2023

Acknowledgements

First and foremost, I want to thank Federico for always supporting me in every way and tolerating my nocturnal lifestyle during the hardest parts of the past four and a half years. I am not sure how I could have done it without you.

I would also like to thank Raphael Hauser for introducing me to integer programming through a very well presented lecture series in my masters year at Oxford. I am also very grateful to Raphael for suggesting the topic of Boolean factor analysis for my master thesis which sowed the seeds of this PhD thesis. Furthermore, I would also like to thank Raphael for helping me secure the generous support of the Alan Turing Institute (and many thanks to the Alan Turing Institute too!) and for the continued support over the years of the PhD.

I would also like to thank Oktay Günlük for teaching me so much about integer programming, for introducing me to people in the integer and combinatorial optimisation community (including Ahmad), for suggesting to try the column generation approach, for pushing me to submit my writeups even when I was not sure they were good enough and for always being so fast in responding and helpful during our meetings.

I would like to thank Ahmad Abdi for teaching the combinatorial optimisation course at LTCC, which was the first taught combinatorial optimisation course I took and greatly enjoyed, and it gave me confidence to do research in the area. I am also immensely grateful to Ahmad for helping me begin my research into minimally non-firm matrices and for all invaluable comments during our discussions.

I would also like to thank Wikipedia, google search and all researchers and lecturers who publish well written books, lecture notes and papers with good examples, as I found such resources vital for my learning.

Finally, I thank my examiners Colin McDiarmid and Giacomo Zambelli for their careful reading, valuable comments and constructive corrections.

Abstract

For a binary matrix $\mathbf{X}$, the Boolean rank $\mathfrak{br}(\mathbf{X})$ is the smallest integer for which $\mathbf{X}$ can be factorised into the Boolean matrix product of two binary matrices $\mathbf{A}$ and $\mathbf{B}$ with inner dimension $\mathfrak{br}(\mathbf{X})$. The isolation number $\mathfrak{i}(\mathbf{X})$ of $\mathbf{X}$ is the maximum number of 1s no two of which are in a same row, column or a 2×2 submatrix of all 1s.

In Part I. of this thesis, we continue Anna Lubiw's study [77] of firm matrices. $\mathbf{X}$ is said to be firm if $\mathfrak{i}(\mathbf{X}) = \mathfrak{br}(\mathbf{X})$ and this equality holds for all its submatrices. We show that the stronger concept of superfirmness of $\mathbf{X}$ is equivalent to having no odd holes in the rectangle cover graph of $\mathbf{X}$, the graph in which $\mathfrak{br}(\mathbf{X})$ and $\mathfrak{i}(\mathbf{X})$ translate to the clique cover number and the independence number, respectively. A binary matrix is minimally non-firm if it is not firm but all of its proper submatrices are. We introduce a matrix operation that leads to generalised binary matrices and, under some conditions, preserves firmness and superfirmness. Then we use this matrix operation to derive several infinite families of minimally non-firm matrices. To the best of our knowledge, minimally non-firm matrices have not been studied before and our constructions provide the first infinite families of them.

In Part II. of this thesis, we explore rank-k binary matrix factorisation (k-BMF). In k-BMF, we are given an $m \times n$ binary matrix $\mathbf{X}$ with possibly missing entries and need to find two binary matrices $\mathbf{A}$ and $\mathbf{B}$ of dimension $m \times k$ and $k \times n$ respectively, which minimise the distance between $\mathbf{X}$ and the Boolean matrix product of $\mathbf{A}$ and $\mathbf{B}$ in the squared Frobenius norm. We present a compact and two exponential size integer programs (IPs) for k-BMF and show that the compact IP has a weak LP relaxation, while the exponential size IPs have a stronger equivalent LP relaxation. We introduce a new objective function, which differs from the traditional squared Frobenius objective in attributing a weight to zero entries of the input matrix that is proportional to the number of times a zero is erroneously covered in a rank-k factorisation. For one of the exponential size IPs we describe a computational approach based on column generation. Experimental results on synthetic and real word datasets suggest that our integer programming approach is competitive against available methods for k-BMF and provides accurate low-error factorisations.

Statement of Originality

This thesis is entirely my own work except where otherwise indicated.

Some of the results from Part I. are published in a conference paper, authored solely by me, in the Proceedings of ISCO 2022 [60]. In the future, I may try to submit an extended journal paper about Part I.

Part II. is based on an a paper [49] accepted into the journal of Mathematics of Operations Research before the completion of this thesis and published afterwards. The work in this paper is written by me and co-authored with my supervisors Oktay and Raphael and is an extended version of a conference paper that is published in the Proceedings of AAAI 2021 [62]. The code relating to these two papers can be found on my github page [59]. Both of these papers, and Part II. Section 9.1 of this thesis, state a compact integer program for k-BMF which first appeared in my MSc thesis in 2017 and then in a short (4-page) NeurIPS 2017 workshop paper [61] co-authored by Oktay, Raphael and me, and published before the start of my PhD. While the compact integer program is stated in these two previous works, it is only computationally explored there and we do new analysis about it in papers [62], [49] and Chapter 9.

Contents

Chapter 1

Introduction

This thesis is about two problems related to factorising binary matrices under Boolean arithmetic. Boolean arithmetic on binary numbers $\{0, 1\}$ interprets 1s as ’True’ and 0s as ’False’ and uses logical disjunction ($\vee$) as *Boolean addition* and logical conjunction ($\wedge$) as *Boolean multiplication*. Boolean multiplication on binary numbers $a, b \in \{0, 1\}$ coincides with standard multiplication, hence we can simply write $a\, b$ in place of $a \wedge b$. Boolean addition however, is different from standard addition as it obeys the law $1 \vee 1 = 1$, which is usually called Boolean non-linearity.

Boolean arithmetic can be extended to binary matrices. The Boolean matrix product of two binary matrices $\mathbf{A} \in \{0, 1\}^{m \times k}$ and $\mathbf{B} \in \{0, 1\}^{k \times n}$, denoted by $\mathbf{A} \circ \mathbf{B}$, is equal to the binary matrix $\mathbf{X} \in \{0, 1\}^{m \times n}$ whose entries are given by

$$x_{i,j} = \bigvee_{\ell=1}^{k} (a_{i,\ell}\, b_{\ell,j}).$$

The *Boolean rank* [55, Definition 1.4.2] of a binary matrix $\mathbf{X}$ is the smallest integer $\mathfrak{br}(\mathbf{X})$ for which there exist binary matrices $\mathbf{A}$ and $\mathbf{B}$ with inner dimension $\mathfrak{br}(\mathbf{X})$ such that $\mathbf{X} = \mathbf{A} \circ \mathbf{B}$. For instance, the matrix below (empty entries corresponding to 0s) has an exact binary matrix factorisation under Boolean arithmetic with inner dimension 2 as shown below, hence its Boolean rank is at most 2,

$$\begin{bmatrix} 1 & 1 & \\ 1 & 1 & 1 \\ & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \\ 1 & 1 \\ & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 & \\ & 1 & 1 \end{bmatrix}.$$

The two problems that we explore in this thesis are both related to the concept of Boolean rank. In Part I, we study some problems related to exact binary matrix factorisation. More specifically, we study binary matrices for which some special relation holds between the Boolean rank and its weak dual, the isolation number. An *isolated set* of a binary matrix is a subset of its 1s no two of which are in a same row, column or a 2×2 submatrix of all 1s. The *isolation number* $\mathfrak{i}(\mathbf{X})$ of a binary matrix $\mathbf{X}$ is the cardinality of a maximum isolated set. For instance, the below matrix has isolation number 2 and several maximum isolated sets, two of which are indicated,

$$\begin{bmatrix} \colorbox{lightgray}{1} & 1 & \\ 1 & 1 & 1 \\ & 1 & \colorbox{lightgray}{1} \end{bmatrix}, \qquad \begin{bmatrix} 1 & \colorbox{lightgray}{1} & \\ 1 & 1 & \colorbox{lightgray}{1} \\ & 1 & 1 \end{bmatrix}.$$

It can be shown that the isolation number is always less than or equal to the Boolean rank for all binary matrices. A binary matrix $\mathbf{X}$ is said to be *firm* [77] if $\mathfrak{i}(\mathbf{X}) = \mathfrak{br}(\mathbf{X})$ and $\mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}')$ also holds for all submatrices $\mathbf{X}'$ of $\mathbf{X}$. In Part I, we explore firm matrices and try to study them through minimal counterexamples. We define a binary matrix $\mathbf{X}$ to be *minimally non-firm* if $\mathbf{X}$ has $\mathfrak{i}(\mathbf{X}) < \mathfrak{br}(\mathbf{X})$ but for all *proper* submatrices $\mathbf{X}'$ of $\mathbf{X}$ we have $\mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}')$. Our main contribution in Part I. is to list several infinite families of minimally non-firm matrices.

In Part II, we look at a problem called rank-k binary matrix factorisation (k-BMF). In this problem we are given an input binary matrix $\mathbf{X}$ and a small positive integer k and need to compute a binary matrix $\mathbf{Z}$ of Boolean rank at most k which best approximates the input matrix $\mathbf{X}$. We approach this problem through integer programming. We give and analyse three integer programming formulations for k-BMF. Then we present a computational approach based on column generation and explore its applicability on artificial and real world datasets.

In the remainder of this chapter, we give an in depth introduction to the Boolean rank, isolation number and all the concepts that are useful for exact and rank-k binary matrix factorisation. In particular, we explore both problems in a graph theory setting and give a detailed account on their complexity status. At the beginning of both parts of thesis, we will state our detailed contributions and the layout for the part.

1.1 Boolean rank and rectangles

Let $\mathbf{X} \in \{0,1\}^{m \times n}$. The *Boolean rank* of $\mathbf{X}$ is the smallest integer $\mathfrak{br}(\mathbf{X})$ for which there exist matrices $\mathbf{A} \in \{0,1\}^{m \times \mathfrak{br}(\mathbf{X})}$ and $\mathbf{B} \in \{0,1\}^{\mathfrak{br}(\mathbf{X}) \times n}$ such that $\mathbf{X} = \mathbf{A} \circ \mathbf{B}$ [55, Definition 1.4.2]. A rank-1 binary matrix is of the form $\boldsymbol{a}\boldsymbol{b}^\top$ for some non-zero binary vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and is often referred to as a *rectangle matrix* [21, pg 178]. The definition of Boolean rank implies that $\mathbf{X}$ can be decomposed as the Boolean sum of

$\mathfrak{br}(\mathbf{X})$ rank-1 binary matrices,

$$\mathbf{X} = \bigvee_{\ell=1}^{\mathfrak{br}(\mathbf{X})} \boldsymbol{a}_\ell \boldsymbol{b}_\ell^\top,$$

where $\boldsymbol{a}_\ell$ are columns of $\mathbf{A}$ and $\boldsymbol{b}_\ell^\top$ are rows of $\mathbf{B}$. For example, the Boolean rank-2 matrix below is the Boolean sum of two rectangle matrices,

$$\begin{bmatrix} 1 & 1 & \\ 1 & 1 & 1 \\ & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \\ 1 & 1 & \\ & & \end{bmatrix} \vee \begin{bmatrix} & & \\ & 1 & 1 \\ & 1 & 1 \end{bmatrix}. \tag{1.1.1}$$

From the definition, a few immediate properties of the Boolean rank follow. Let $\mathbf{I}_n$ be the $n \times n$ identity matrix. Taking $\mathbf{A} = \mathbf{I}_m$ and $\mathbf{B} = \mathbf{X}$ or $\mathbf{A} = \mathbf{X}$ and $\mathbf{B} = \mathbf{I}_n$ shows that $\mathfrak{br}(\mathbf{X}) \leq \min\{m, n\}$. In addition, the Boolean rank is invariant under transposition, permutation of rows and columns, and appending duplicate rows or columns to $\mathbf{X}$. The invariance under row and column duplication holds because if a row of $\mathbf{X}$ is duplicated, it suffices to duplicate the corresponding row of $\mathbf{A}$ in any factorisation of $\mathbf{X}$. Due to this, we may consider only matrices that have no row and no column duplicates. Furthermore, for any $\mathbf{Y} \in \{0,1\}^{n \times t}$, we have $\mathfrak{br}(\mathbf{X} \circ \mathbf{Y}) \leq \min\{\mathfrak{br}(\mathbf{X}), \mathfrak{br}(\mathbf{Y})\}$ as, for instance $\mathbf{A} \circ (\mathbf{B} \circ \mathbf{Y})$ is a valid factorisation of $\mathbf{X} \circ \mathbf{Y}$.

For some positive integer m let $[m] := \{1, 2, \ldots, m\}$. We define the support of the 1s and 0s of a binary matrix $\mathbf{X} \in \{0,1\}^{m \times n}$ by

$$\begin{aligned} \operatorname{supp}_1(\mathbf{X}) &= \{(i,j) \in [m] \times [n] : x_{i,j} = 1\}, \\ \operatorname{supp}_0(\mathbf{X}) &= \{(i,j) \in [m] \times [n] : x_{i,j} = 0\}. \end{aligned}$$

Let $I \subseteq [m]$ and $J \subseteq [n]$ be a subset of row and column indices of $\mathbf{X}$, respectively. A *submatrix* of $\mathbf{X}$ identified by $I \times J$ is obtained by deleting the rows not in I and the columns not in J. If $I \subsetneq [m]$ or $J \subsetneq [n]$ then $I \times J$ is a *proper submatrix* of $\mathbf{X}$. A submatrix of $\mathbf{X}$ identified by $I \times J$ is called a *rectangle* if $I \times J \subseteq \operatorname{supp}_1(\mathbf{X})$. In this case, we may simply say that $I \times J$ is a rectangle of $\mathbf{X}$. Observe, that if $I \times J$ is a rectangle of $\mathbf{X}$, then the rank-1 binary matrix $\boldsymbol{a}\boldsymbol{b}^\top$ with $a_i = b_j = 1$ for $i \in I, j \in J$ satisfies $\boldsymbol{a}\boldsymbol{b}^\top \leq \mathbf{X}$, where $\leq$ is always understood entry-wise for matrices. $I \times J$ is called a *maximal rectangle* of $\mathbf{X}$ if it is not contained in any other rectangle of $\mathbf{X}$. Let $\mathcal{R}(\mathbf{X})$ denote all the rectangles of $\mathbf{X}$ and let $\mathcal{R}_{\max}(\mathbf{X})$ denote the set of maximal rectangles of $\mathbf{X}$.

In terms of rectangles, one may give an equivalent combinatorial definition of the Boolean rank: The Boolean rank of a binary matrix $\mathbf{X}$ is the minimum number of rectangles needed to cover the 1s of $\mathbf{X}$. This is the reason that in the literature the Boolean rank is often also referred to as the *rectangle cover number*. In fact, one need only to consider maximal rectangles, hence it can be assumed that a minimum rectangle cover of $\mathbf{X}$ consists only of maximal rectangles. For instance, our toy matrix in Equation (1.1.1) may be covered by two maximal rectangles $\{1,2\} \times \{1,2\}$ and $\{2,3\} \times \{2,3\}$ that correspond to the rectangle matrices given earlier.

A related problem to computing the Boolean rank, is to find the largest rectangle of $\mathbf{X}$. In the *maximum rectangle* problem a rectangle of $\mathbf{X}$ with the maximum number of 1s is sought. Let $\mathfrak{mr}(\mathbf{X})$ denote the number of 1s in a maximum rectangle of $\mathbf{X}$. As any rectangle of $\mathbf{X}$ has size at most $\mathfrak{mr}(\mathbf{X})$, we need at least $|\operatorname{supp}_1(\mathbf{X})|/\mathfrak{mr}(\mathbf{X})$ rectangles to cover all 1s of $\mathbf{X}$. Therefore, the following simple bound holds on $\mathfrak{br}(\mathbf{X})$:

$$\left\lceil \frac{|\operatorname{supp}_1(\mathbf{X})|}{\mathfrak{mr}(\mathbf{X})} \right\rceil \leq \mathfrak{br}(\mathbf{X}). \tag{1.1.2}$$

1.1.1 Boolean row and column rank

Considering the Boolean rank of binary matrices, it is natural to define similar concepts to some basic real linear algebra in the Boolean setting. We give an introduction to Boolean linear algebra as it is defined in the book of Kim [55].

A *Boolean subspace* of $\{0,1\}^n$ is a subset of $\{0,1\}^n$ containing the zero vector and closed under Boolean addition ($\vee$). The *Boolean span* of k binary vectors $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\} \subseteq \{0,1\}^n$ is the set of all binary vectors that can be expressed as the Boolean sum of vectors in W using binary coefficients,

$$\operatorname{span}(W) = \{\boldsymbol{a} \in \{0,1\}^n : \boldsymbol{a} = \bigvee_{\ell=1}^{k} \alpha_\ell \boldsymbol{w}_\ell, \alpha_\ell \in \{0,1\}\}.$$

We say $\boldsymbol{b} \in \{0,1\}^n$ is *Boolean dependent* on W if $\boldsymbol{b} \in \operatorname{span}(W)$ and *Boolean independent* from W if $\boldsymbol{b} \notin \operatorname{span}(W)$. The set of k vectors W is said to be a *Boolean independent set* if for all $\ell \in [k]$ $\boldsymbol{w}_\ell \in W$ is Boolean independent from $W \setminus \{\boldsymbol{w}_\ell\}$, i.e. none of the vectors $\boldsymbol{w}_\ell$ can be expressed as the Boolean sum of the other vectors in W. A subset B of a Boolean subspace S is called a *Boolean basis* of S if B is a Boolean independent set and $\operatorname{span}(B) = S$. If S is a Boolean subspace, then there exists a unique subset B of S which is the basis of S [55, Theorem 1.1.1]. We denote this unique basis of S by $\operatorname{basis}(S)$.

Similarly to real matrices we may define row and column spaces of binary matrices under the Boolean arithmetic setting. The *Boolean row space* (or *Boolean column space*) of $\mathbf{X} \in \{0,1\}^{m \times n}$ is defined as the Boolean span of the row (column) vectors of $\mathbf{X}$,

$$\begin{aligned} BRS(\mathbf{X}) &= \operatorname{span}(\{\mathbf{X}_{i,:} : i \in [m]\}) \subseteq \{0,1\}^n \\ BCS(\mathbf{X}) &= \operatorname{span}(\{\mathbf{X}_{:,j} : j \in [n]\}) \subseteq \{0,1\}^m, \end{aligned}$$

where $\mathbf{X}_{i,:}$ and $\mathbf{X}_{:,j}$ denote the i-th row and the j-th column of $\mathbf{X}$, respectively. It can be proved [55, Theorem 1.2.3] that the Boolean row and column space have the same cardinality for all binary matrices,

$$|BRS(\mathbf{X})| = |BCS(\mathbf{X})|.$$

The *Boolean row and column ranks* of $\mathbf{X}$ are defined to be the cardinality of the unique Boolean basis of $BRS(\mathbf{X})$ and $BCS(\mathbf{X})$, respectively [55, Definition 1.2.15]. Equivalently, the Boolean row (column) rank of $\mathbf{X}$ is the maximum number of rows (columns) of $\mathbf{X}$ that form a Boolean independent set. The Boolean column and row rank are not necessarily equal as shown in [55]. Let $\mathbf{X}$ be the binary matrix in Equation (1.1.3) below. $\mathbf{X}$ has full Boolean column rank of 4, but it only has Boolean row rank 3 as its second row is the Boolean sum of the first and third rows.

$$\begin{bmatrix} 1 & 1 & & \\ 1 & 1 & 1 & \\ 1 & & 1 & \\ & 1 & 1 & 1 \end{bmatrix} \tag{1.1.3}$$

Furthermore, the Boolean row and column ranks usually differ from the Boolean rank that we have seen earlier, which is also called *Boolean factor rank*. Let $\mathbf{X}$ be the binary matrix in Equation (1.1.4) below. $\mathbf{X}$ has Boolean row and column rank equal to 4 while the Boolean factor rank of it is 3.

$$\begin{bmatrix} 1 & 1 & & \\ 1 & 1 & 1 & \\ & 1 & 1 & 1 \\ & & 1 & 1 \end{bmatrix} \tag{1.1.4}$$

In fact, an equivalent definition of the Boolean factor rank can be given as the least integer $\mathfrak{br}(\mathbf{X})$ such that $BCS(\mathbf{X})$ (or equivalently, $BRS(\mathbf{X})$) is contained in a space spanned by $\mathfrak{br}(\mathbf{X})$ vectors [55, pg. 38].

If a binary matrix $\mathbf{X} \in \{0,1\}^{m \times n}$ has Boolean row rank r then each row of $\mathbf{X}$ can be written as the Boolean sum of r rows of $\mathbf{X}$. Therefore we have, $\mathbf{X} = \mathbf{A} \circ \mathbf{B}$, where $\mathbf{B}$ is $r \times n$ binary matrix that contains the r rows that form the Boolean basis of the Boolean row space of $\mathbf{X}$ and $\mathbf{A}$ is the $m \times r$ matrix which contains the coefficients of expressing each row of $\mathbf{X}$ as the Boolean sum of rows of $\mathbf{B}$. This form shows that for any binary matrix the Boolean factor rank is less than or equal to the Boolean row rank and the Boolean column rank [55, Theorem 1.4.1],

$$\mathfrak{br}(\mathbf{X}) \leq \min\left\{|\text{basis}(BRS(\mathbf{X}))|,\ |\text{basis}(BCS(\mathbf{X}))|\right\}. \tag{1.1.5}$$

A lower bound on $\mathfrak{br}(\mathbf{X})$ presented in [94, Theorem 8.2] may also be obtained using Boolean row and column spaces. Let $\mathbf{X} = \mathbf{A} \circ \mathbf{B}$ be an optimal factorisation of $\mathbf{X} \in \{0,1\}^{m \times n}$ with inner dimension of $\mathfrak{br}(\mathbf{X})$. Any vector $\boldsymbol{y}$ in the Boolean column space of $\mathbf{X}$, $\boldsymbol{y} \in BCS(\mathbf{X})$ can be written as $\boldsymbol{y} = \mathbf{X} \circ \boldsymbol{v}$ for some $\boldsymbol{v} \in \{0,1\}^n$. Using the optimal factorisation of $\mathbf{X}$, we can then write $\boldsymbol{y} = \mathbf{A} \circ \boldsymbol{w}$ where $\mathbf{B} \circ \boldsymbol{v} = \boldsymbol{w} \in \{0,1\}^{\mathfrak{br}(\mathbf{X})}$. Since this holds for any $\boldsymbol{y} \in BCS(\mathbf{X})$, and there are at most $2^{\mathfrak{br}(\mathbf{X})}$ many choices for the binary vector $\boldsymbol{w}$, we have $|BCS(\mathbf{X})| \leq 2^{\mathfrak{br}(\mathbf{X})}$. Therefore, we have the following lower bound on the Boolean factor rank,

$$\lceil \log |BCS(\mathbf{X})| \rceil \leq \mathfrak{br}(\mathbf{X}), \tag{1.1.6}$$

where log denotes the base 2 logarithm.

This technique can also be used to prove another lower bound for matrices that have no repeated columns. So let $\mathbf{X}$ not have any repeated columns, and let $\mathbf{X} = \mathbf{A} \circ \mathbf{B}$ be an optimal factorisation. For each $j \in [n]$, the j-th column of $\mathbf{X}$ may be written as $\mathbf{X}_{:,j} = \mathbf{X} \circ \boldsymbol{e}_j = (\mathbf{A} \circ \mathbf{B}) \circ \boldsymbol{e}_j = \mathbf{A} \circ \mathbf{B}_{:,j}$, where $\boldsymbol{e}_j$ is the j-th standard unit column vector of appropriate dimension. This shows that $\mathbf{B}$ cannot have any repeated columns either. As there are $2^{\mathfrak{br}(\mathbf{X})}$ possibilities for a column of $\mathbf{B}$, we have $n \leq 2^{\mathfrak{br}(\mathbf{X})}$. This reasoning clearly also holds for matrices that have no repeated rows. Hence if $\mathbf{X}$ has no repeated rows nor repeated columns then,

$$\max\{\lceil \log m \rceil, \lceil \log n \rceil\} \leq \mathfrak{br}(\mathbf{X}). \tag{1.1.7}$$

This lower bound can be further strengthened for a smaller class of matrices in which rows are pairwise incomparable. Let $\mathbf{X}$ be called a *row-clutter* matrix if for any distinct $i, \ell \in [m]$ we have $\mathbf{X}_{i,:} \not\leq \mathbf{X}_{\ell,:}$. The name comes from the definition of clutters. A *clutter* is a family of subsets $\mathcal{F}$ on a finite ground set such that for any two distinct sets $F, G \in \mathcal{F}$ we have $F \subsetneq G$. Sperner's lemma shows that a clutter on an n element ground set can have size at most $\binom{n}{\lfloor n/2 \rfloor}$. Hence if $\mathbf{X} \in \{0,1\}^{m \times n}$ is a row-clutter matrix, then $m \leq \binom{n}{\lfloor n/2 \rfloor}$. Furthermore, in any optimal factorisation $\mathbf{X} = \mathbf{A} \circ \mathbf{B}$, $\mathbf{A}$ must be a row clutter matrix as well as $\mathbf{X}_{i,:} = \mathbf{A}_{i,:} \circ \mathbf{B}$. Therefore, $m \leq \binom{\mathfrak{br}(\mathbf{X})}{\lfloor \mathfrak{br}\mathbf{X}/2 \rfloor}$ and we obtain the following lower bound on row clutter matrices which was first proved by Caen et al. [28],

$$s(m) := \min\left\{p : m \leq \binom{p}{\lfloor p/2 \rfloor}\right\} \leq \mathfrak{br}(\mathbf{X}). \tag{1.1.8}$$

1.1.2 Relation to other matrix ranks

A related concept to the Boolean rank is the integer rank or the rectangle partition number. The *integer rank* of a binary matrix $\mathbf{X} \in \{0,1\}^{m \times n}$ is the smallest integer r for which there exist matrices $\mathbf{A} \in \{0,1\}^{m \times r}$ and $\mathbf{B} \in \{0,1\}^{r \times n}$ such that $\mathbf{X} = \mathbf{AB}$, using ordinary matrix multiplication. In terms of rectangles, the integer rank is equal to the minimum number of disjoint rectangles needed to cover the 1s of $\mathbf{X}$. It is easy to see that the Boolean rank is always less than or equal to the integer rank.

Much theoretical interest in the Boolean rank is driven by the fact that it provides a lower bound on the nonnegative rank. The *nonnegative rank* of a nonnegative matrix $\mathbf{Y} \in \mathbb{R}_+^{m \times n}$ is the smallest integer t for which there exist nonnegative matrices $\mathbf{W} \in \mathbb{R}_+^{m \times t}$ and $\mathbf{H} \in \mathbb{R}_+^{t \times n}$ for which $\mathbf{Y} = \mathbf{WH}$, under standard matrix multiplication. It can be readily checked that if $\mathbf{X}$ is the binary matrix which has a 1 in the position of every nonzero entry of a nonnegative matrix $\mathbf{Y}$ and 0 otherwise, then the Boolean rank of $\mathbf{X}$ is a lower bound on the nonnegative rank of $\mathbf{Y}$.

This property makes the Boolean rank and any lower bound on it a powerful tool in the field of linear extension complexity of polytopes [106]. In extension complexity, the relationship between facets and vertices of a polytope is encoded in a slack matrix which is non-negative, and the nonnegative rank of the slack matrix equals the minimum size of an extended formulation of the polytope. Further applications of the Boolean rank can be found in the field of communication complexity. In that context, the ceiling of the base 2 logarithm of the Boolean rank of the binary communication matrix of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ equals the non-deterministic communication complexity of the function [107].

Interestingly, the Boolean rank does not have a clear relationship to the standard real rank as one can find examples in which the Boolean rank is strictly less or strictly greater than the real rank. For instance, the matrix on the left hand side below has Boolean rank 2 and real rank 3 while the matrix on the right hand side has real rank 3 and Boolean rank 4,

$$\begin{bmatrix} 1 & 1 & \\ 1 & 1 & 1 \\ & 1 & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \\ 1 & & & 1 \end{bmatrix}.$$

1.2 Isolation number

A frequently useful lower bound on $\mathfrak{br}(\mathbf{X})$ is defined in 1983 by Gregory et al. [44]. An *isolated set* of $\mathbf{X}$ is a set $S \subseteq \operatorname{supp}_1(\mathbf{X})$ such that for all distinct (i_1, j_1), (i_2, j_2)

in S we have

(1) $i_1 \neq i_2$ and $j_1 \neq j_2$, and

(2) $(i_1, j_2) \in \mathrm{supp}_0(\mathbf{X})$ or $(i_2, j_1) \in \mathrm{supp}_0(\mathbf{X})$ or both.

The cardinality of a maximum isolated set in $\mathbf{X}$ is called the *isolation number* of $\mathbf{X}$ and denoted by $\mathfrak{i}(\mathbf{X})$ [44]. In the field of communication complexity, $\mathfrak{i}(\mathbf{X})$ is often referred to as the fooling set bound [66].

If $\mathbf{X}$ has isolation number $\mathfrak{i}(\mathbf{X})$, then there are $\mathfrak{i}(\mathbf{X})$ 1s of $\mathbf{X}$ that cannot be contained in a common rectangle, hence at least $\mathfrak{i}(\mathbf{X})$ rectangles are needed to cover the 1s of $\mathbf{X}$. Therefore, the isolation number provides a lower bound on the Boolean rank. In many cases however, the inequality between the Boolean rank and isolation number may be strict.

Example 1.2.1. *The most well known example where the isolation number and the Boolean rank are not equal is the complement of the identity matrix. This matrix is defined as $\overline{\mathbf{I}}_n := \mathbf{J}_n - \mathbf{I}_n$, where $\mathbf{J}_n$ is the $n \times n$ matrix of all* 1*s and $\mathbf{I}_n$ is the identity matrix. Since $\overline{\mathbf{I}}_n$ is a row-clutter matrix (and also column-clutter) we have $s(n) \leq \mathfrak{br}(\overline{\mathbf{I}}_n)$, where $s(n)$ is defined in Equation* (1.1.8). *In [28] (where the row-clutter matrix bound is proved), a feasible factorisation of inner dimension $s(n)$ is given for $\overline{\mathbf{I}}_n$, hence we have*

$$\mathfrak{br}(\overline{\mathbf{I}}_n) = s(n) = \min\left\{p : n \leq \binom{p}{\lfloor p/2 \rfloor}\right\} \sim \log(n).$$

Observing that for $(i,j) \in \mathrm{supp}_1(\overline{\mathbf{I}}_n)$ to belong to an isolated set of size k, the number of 0*s in row i together with the number of* 0*s in column j has to be at least $k-1$. Any* 1 *of $\overline{\mathbf{I}}_n$ has exactly one* 0 *in its row and one* 0 *in its column, therefore it can belong to an isolated set of size at most* 3. *Thus we have $\mathfrak{i}(\overline{\mathbf{I}}_n) \leq 3$ (which seems to be a folklore result for which we could not find a clear earliest reference). For $n = 1, 2$ we have $\mathfrak{i}(\overline{\mathbf{I}}_1) = 0$ and $\mathfrak{i}(\overline{\mathbf{I}}_2) = 2$. On the other hand, for $n \geq 3$, it is easy to find an isolated set of size* 3, *hence we have*

$$\mathfrak{i}(\overline{\mathbf{I}}_n) = 3 \qquad \textit{for all } \ n \geq 3.$$

The above example gives an idea for a simple upper bound on the isolation number based on the number of the 0s of the matrix. For each $(i,j) \in \mathrm{supp}_1(\mathbf{X})$, we have that any isolated S containing (i,j) satisfies,

$$(i,j) \in S \implies |S| \leq |\mathrm{supp}_0(\mathbf{X}_{i,:})| + |\mathrm{supp}_0(\mathbf{X}_{:,j})| + 1. \tag{1.2.1}$$

A similar global bound may be provided on $\mathfrak{i}(\mathbf{X})$. If S is an isolated set of $\mathbf{X}$, then by definition for any two distinct elements $(i_1, j_1), (i_2, j_2) \in S$ we have at least one of (i_1, j_2) or (i_2, j_1) in $\mathrm{supp}_0(\mathbf{X})$. Therefore, for $\mathbf{X}$ to have an isolated set of size t it has to have at least $\binom{t}{2}$ zeros and we have

$$\mathfrak{i}(\mathbf{X}) \leq \max \left\{ p : \binom{p}{2} \leq |\,\mathrm{supp}_0(\mathbf{X})| \right\}. \tag{1.2.2}$$

A weak dual problem of the maximal rectangle problem may be considered in terms of isolated sets that we mention for the sake of completeness. This problem asks to cover the 1s of $\mathbf{X}$ with a minimum number of isolated sets. We are not aware of any results on this problem.

1.3 Generalised binary matrices

A *generalised binary matrix* is a matrix $\mathbf{X}$ over $\{0, 1, ?\}$ [77]. The ? entries are considered to be unknown, missing or 'no care' elements. The importance of the new entry type ? is that these entries may be used to form rectangles but need not be covered in a feasible factorisation of $\mathbf{X}$.

Let $\mathrm{supp}_1(\mathbf{X})$ contain the indices of 1s of $\mathbf{X}$ just as in the case of *standard* binary matrices ($\{0, 1\}$-matrices), $\mathrm{supp}_0(\mathbf{X})$ the indices of 0s and $\mathrm{supp}_?(\mathbf{X})$ the indices of ?s,

$$\mathrm{supp}_?(\mathbf{X}) := \{(i, j) : x_{i,j} = ?\}.$$

Let $\Omega(\mathbf{X})$ denote the set of indices of all known entries of $\mathbf{X}$,

$$\Omega(\mathbf{X}) = \mathrm{supp}_0(\mathbf{X}) \cup \mathrm{supp}_1(\mathbf{X}).$$

A *rectangle* of a generalised binary matrix $\mathbf{X}$ is a submatrix $I \times J$ which satisfies $I \times J \subseteq (\mathrm{supp}_1(\mathbf{X}) \cup \mathrm{supp}_?(\mathbf{X}))$. Thus a rectangle of $\mathbf{X}$ may be any submatrix that does not contain 0s. An *isolated set* S of $\mathbf{X}$ is a subset of $\mathrm{supp}_1(\mathbf{X})$, no two of which can be contained in a common rectangle. Similarly to standard binary matrices, we denote the isolation number, the cardinality of a maximum isolated set by $\mathfrak{i}(\mathbf{X})$ and the Boolean rank, the minimum number of rectangles needed to cover $\mathrm{supp}_1(\mathbf{X})$ by $\mathfrak{br}(\mathbf{X})$.

Now let $\mathbf{X}$ be a standard binary matrix. For a non-empty set $P \subset \mathrm{supp}_1(\mathbf{X})$, let $\mathbf{X}^P$ denote the generalised binary matrix obtained from $\mathbf{X}$ by replacing the 1s in P by ?s, so $\mathrm{supp}_1(\mathbf{X}^P) = \mathrm{supp}_1(\mathbf{X}) \setminus P$ and $\mathrm{supp}_?(\mathbf{X}^P) = P$. Since ?s need not be covered and cannot be members of an isolated set, for any $P \subset \mathrm{supp}_1(\mathbf{X})$ we have

$$\mathfrak{br}(\mathbf{X}^P) \leq \mathfrak{br}(\mathbf{X}), \qquad\qquad \mathfrak{i}(\mathbf{X}^P) \leq \mathfrak{i}(\mathbf{X}). \tag{1.3.1}$$

A maximum rectangle of a generalised binary matrix is one that contains the maximum number of 1s. Let $\mathfrak{mr}(\mathbf{X}^P)$ denote the number of 1s in a maximum rectangle of $\mathbf{X}^P$. The maximum-rectangle Boolean rank bound given in Equation (1.1.2) may be straightforwardly extended to generalised binary matrices to get

$$\left\lceil \frac{|\operatorname{supp}_1(\mathbf{X}^P)|}{\mathfrak{mr}(\mathbf{X}^P)} \right\rceil \leq \mathfrak{br}(\mathbf{X}^P). \tag{1.3.2}$$

1.4 Rank-k binary matrix factorisation

In the second part of this thesis, we will consider the rank-k binary matrix factorisation (k-BMF) problem: we are given an $m \times n$ generalised binary matrix $\mathbf{X}$ and a small positive integer k and need to find two binary matrices $\mathbf{A}$ and $\mathbf{B}$ of dimension $m \times k$ and $k \times n$, respectively, which minimise the distance between $\mathbf{X}$ and the Boolean product of $\mathbf{A}$ and $\mathbf{B}$ in the squared Frobenius distance. Rank-k binary matrix factorisation was first defined in [84] by Miettinen et al. The motivation behind the definition of k-BMF was to devise a method that is able to extract k hidden features from datasets which can be represented by generalised binary matrices. In many data science applications, data is contained in a generalised binary matrix where the ?'s denote missing entries while 1's and 0's encode the answers to yes-or-no questions. In these applications, the data matrix $\mathbf{X}$ is usually large and may contain erroneous entries as well. So rather then finding the Boolean rank of $\mathbf{X}$, it is more useful to approximate $\mathbf{X}$ with a binary matrix of fixed small Boolean rank k which then can give some insight into what the underlying patterns in the data are and also provide a completion of missing entries of $\mathbf{X}$. This is done in an analogous way to how singular value decomposition and nonnegative matrix factorisation can be used to find hidden features in real and nonnegative datasets. In Section 1.4 of Part II. we give a detailed account on the practical motivation behind k-BMF and several problems and previous work related to k-BMF.

Formally, in k-BMF we are given an input matrix $\mathbf{X} \in \{0, 1, ?\}^{m \times n}$ and an input integer $k \in \mathbb{Z}_{++} := \{1, 2, \dots\}$ such that $k \ll \min\{m, n\}$, and need to find two binary matrices $\mathbf{A} \in \{0, 1\}^{m \times k}$, $\mathbf{B} \in \{0, 1\}^{k \times n}$, so that we minimise the squared Frobenius distance between $\mathbf{X}$ and $\mathbf{A} \circ \mathbf{B}$. Therefore, we aim to solve

$$\zeta(\mathbf{X}, k) = \min\{\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{A} \circ \mathbf{B})\|_F^2 : \mathbf{A} \in \{0, 1\}^{m \times k}, \mathbf{B} \in \{0, 1\}^{k \times n}\},$$

where $\mathcal{P}_\Omega$ is the projection onto the space of known entries, so the error is evaluated only over the known entries of $\mathbf{X}$, $(i, j) \in \Omega(\mathbf{X})$; and $\|\mathbf{Y}\|_F$ denotes the Frobenius norm defined by $\sqrt{\sum_i \sum_j y_{i,j}^2}$. Hence, we have

$$\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{A} \circ \mathbf{B})\|_F^2 = \sum_{(i,j) \in \Omega(\mathbf{X})} (x_{i,j} - (\mathbf{A} \circ \mathbf{B})_{i,j})^2 .$$

Let $\mathbf{Z} = \mathbf{A} \circ \mathbf{B}$ for some $\mathbf{A} \in \{0,1\}^{m \times k}$ and $\mathbf{B} \in \{0,1\}^{k \times n}$. Then by the fixed inner dimension k of $\mathbf{A} \circ \mathbf{B}$, we have $\mathfrak{br}(\mathbf{Z}) \leq k$ and we call $\mathbf{Z}$ a *rank-k factorisation or completion of* $\mathbf{X}$. Since $\mathbf{X}$ is a generalised binary matrix and $\mathbf{Z}$ is a standard binary matrix, the factorisation error in the squared Frobenius norm is over binary entries which coincides with the error in entry-wise ℓ_1-norm. Hence, we can expand $\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Z})\|_F^2$ to get the following linear expression,

$$\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Z})\|_F^2 = \sum_{(i,j) \in \Omega(\mathbf{X})} |x_{i,j} - z_{i,j}| = \sum_{(i,j) \in \mathrm{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{(i,j) \in \mathrm{supp}_0(\mathbf{X})} z_{i,j} .$$

This form may further be simplified by bringing out the constant term in the objective,

$$\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Z})\|_F^2 = |\,\mathrm{supp}_1(\mathbf{X})| - \sum_{(i,j) \in \mathrm{supp}_1(\mathbf{X})} z_{i,j} + \sum_{(i,j) \in \mathrm{supp}_0(\mathbf{X})} z_{i,j} . \tag{1.4.1}$$

So the rank-k factorisation error $\zeta(\mathbf{X}, k)$ is just the number of 1s of $\mathbf{X}$ not covered, plus the number of 0s of $\mathbf{X}$ erroneously covered by an optimal rank-k factorisation $\mathbf{Z}$ of $\mathbf{X}$.

As $\mathbf{Z}$ is of Boolean rank at most k, it can be written as the Boolean sum of k rank-1 binary matrices, $\mathbf{Z} = \bigvee_{\ell=1}^k \boldsymbol{a}_\ell \boldsymbol{b}_\ell^\top$. Note however, that these rank-1 binary matrices do not necessarily correspond to rectangles of $\mathbf{X}$, as $\mathbf{Z}$ is allowed to cover 0 entries of $\mathbf{X}$. For instance, the optimal rank-1 binary matrix factorisation of the matrix,

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & \\ 1 & 1 & 1 \\ & 1 & 1 \end{bmatrix} ,$$

is given by the 3×3 all 1s matrix $\mathbf{J}_3$ and incurs an error $\zeta(\mathbf{X}, 1) = 2$ by covering the two 0s of $\mathbf{X}$. On the other hand, the largest rectangle of $\mathbf{X}$ is of size 2×2, and using that as a rank-1 factorisation incurs an error of size 3 as 3 1s are not covered. This shows, that one needs to consider all rank-1 binary matrices of the input matrix's dimension as candidates to be included in an optimal rank-k factorisation, and it does not suffice to only consider maximal rectangles of $\mathbf{X}$.

Rank-1 binary matrix factorisation (1-BMF) is closely related to the maximum rectangle problem. In fact, it can be seen as a weighted version of it. In 1-BMF, the optimal solution is of Boolean rank 1, so we may write $\mathbf{Z} = \boldsymbol{a}\boldsymbol{b}^\top$ and

$$\begin{aligned} \zeta(\mathbf{X}, 1) &= \min_{\boldsymbol{a}\in\{0,1\}^m, \boldsymbol{b}\in\{0,1\}^n} \|\mathcal{P}_\Omega(\mathbf{X} - \boldsymbol{a}\boldsymbol{b}^\top)\|_F^2 \\ &= |\operatorname{supp}_1(\mathbf{X})| - \max_{\substack{\boldsymbol{a}\in\{0,1\}^m,\\ \boldsymbol{b}\in\{0,1\}^n}} \left(\sum_{(i,j)\in\operatorname{supp}_1(\mathbf{X})} a_i b_j - \sum_{(i,j)\in\operatorname{supp}_0(\mathbf{X})} a_i b_j \right). \end{aligned} \tag{1.4.2}$$

This form shows that in 1-BMF we are looking for a submatrix of $\mathbf{X}$ which picks up the most number of 1s of $\mathbf{X}$ and the least number of 0s of $\mathbf{X}$. We may define a weight matrix $\boldsymbol{\mathcal{W}} \in \{0, \pm 1\}^{m\times n}$ for $\mathbf{X}$, which has entries given by

$$\mathcal{W}_{i,j} = \begin{cases} +1 & (i,j) \in \operatorname{supp}_1(\mathbf{X}), \\ -1 & (i,j) \in \operatorname{supp}_0(\mathbf{X}), \\ 0 & (i,j) \in \operatorname{supp}_?(\mathbf{X}). \end{cases} \tag{1.4.3}$$

Then 1-BMF is exactly the problem of computing a largest total weight submatrix of $\boldsymbol{\mathcal{W}}$ and setting $a_i = 1$ and $b_j = 1$ for those rows and columns which index such a maximum weight submatrix.

1.5 The bipartite graph setting

All the problems that we have discussed so far in the matrix setting can be equivalently stated as graph problems. Let us review some graph terminology. Let $G = (V, E)$ be a finite undirected graph with vertex set V and edge set E that contains no parallel edges or loops. G is said to be *complete* if every pair of vertices is adjacent. We use K_n to denote the complete graph on n vertices. A *clique* $K \subseteq V$ is a complete subgraph of G. A clique is maximal if it is not contained in any other clique of G. A clique of G is maximum if it has a maximum number of vertices. The cardinality of a maximum clique of G is denoted by $\omega(G)$. The *clique cover number* of G, denoted by $\theta(G)$, is the minimum number of cliques needed to cover the vertices of G. $\theta(G)$ is also called the *clique partition number* as removing overlaps between cliques in a clique cover gives a clique partition of the vertices of equal cardinality.

An *independent (or stable) set* $S \subseteq V$ is a set of pairwise non-adjacent vertices of G. A maximal independent set is not contained in any other independent set of G and a maximum one has the a maximum number of vertices. The *independence or stability number* of G, denoted $\alpha(G)$, is the cardinality of a maximum independent set of G. The *chromatic number* of G, denoted $\chi(G)$ is the minimum number of independent sets needed to partition (or equivalently, cover) the vertices of G. Since a clique and an independent set may intersect at at most one vertex, $\alpha(G) \leq \theta(G)$ and $\omega(G) \leq \chi(G)$. Let $\overline{G}$ be the graph complement of graph G, where two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G. It is easy to see that complements of independent sets are cliques and thus we have $\alpha(G) = \omega(\overline{G})$ and $\theta(G) = \chi(\overline{G})$.

A *bipartite graph* is one whose vertex set can be partitioned into two disjoint independent sets. Let $\mathcal{B}(\mathbf{X})$ be the bipartite graph associated with $\mathbf{X} \in \{0,1\}^{m\times n}$ which has a vertex for each row $i \in [m]$ of $\mathbf{X}$ on one side of the bipartition, a vertex for each column $j \in [n]$ of $\mathbf{X}$ on the other side and an edge (i,j) between vertex i and vertex j if and only if $x_{i,j} = 1$,

$$\mathcal{B}(\mathbf{X}) = ([m], [n], \operatorname{supp}_1(\mathbf{X})).$$

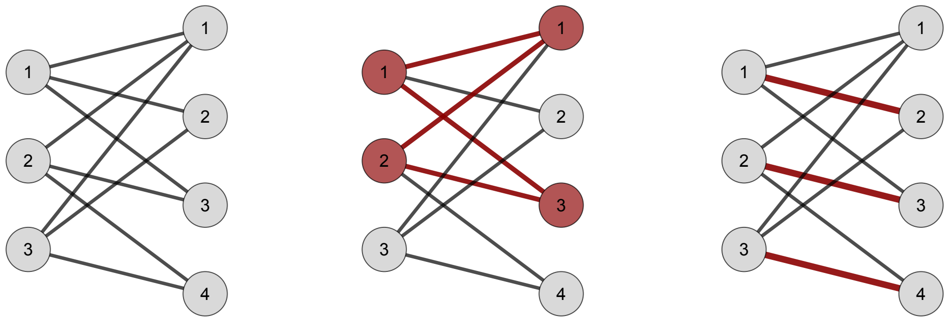
We call $\mathcal{B}(\mathbf{X})$ the *bipartite representation* of $\mathbf{X}$ or $\mathbf{X}$ in the *bipartite setting* and $\mathbf{X}$ is called the *biadjacency matrix* of $\mathcal{B}(\mathbf{X})$. A *complete bipartite graph* is where every vertex on one side of the bipartition is adjacent to every vertex on the other side of the bipartition. The complete bipartite graph with $m+n$ vertices is denoted by $K_{m,n}$. A *biclique* is a complete bipartite subgraph. The minimum number of bicliques needed to cover the *edge set* of a bipartite graph is called the *biclique cover number*. Observe that rectangles of $\mathbf{X}$ and bicliques of $\mathcal{B}(\mathbf{X})$ are in direct correspondence, as a rectangle indexed by $I \times J$ satisfies $I \times J \subseteq \operatorname{supp}_1(\mathbf{X})$ and simply corresponds to the biclique of $\mathcal{B}(\mathbf{X})$ with vertex set $I \cup J$ and edge set $I \times J$. Therefore, $\mathfrak{br}(\mathbf{X})$ is exactly the biclique cover number of $\mathcal{B}(\mathbf{X})$. Similarly, the minimum rectangle partition number or the integer rank of a binary matrix $\mathbf{X}$ is equal to the minimum number of disjoint bicliques needed to cover the edge set of $\mathcal{B}(\mathbf{X})$. Furthermore, a maximum rectangle of $\mathbf{X}$ is then just a biclique of $\mathcal{B}(\mathbf{X})$ with a maximum number of edges, called a *maximum edge biclique*.

A *matching* in a graph is a set of edges that are pairwise non-adjacent, that is they do not share a common vertex. A cycle is said be an *alternating cycle* with respect to a matching if every second edge of the cycle belongs to the matching. We denote a cycle on n vertices by C_n. A matching in which no two edges are contained in an alternating cycle of length four is called an *alternating C_4-free matching*.

Let $S \subseteq \operatorname{supp}_1(\mathbf{X})$ be an isolated set of $\mathbf{X}$. Any two distinct elements $(i_1,j_1),(i_2,j_2)$ in S satisfy $i_1 \neq i_2$ and $j_1 \neq j_2$, which shows that S is a matching in $\mathcal{B}(\mathbf{X})$. Furthermore, the second condition of isolated sets states $(i_1,j_2) \in \operatorname{supp}_0(\mathbf{X})$ or $(i_2,j_1) \in \operatorname{supp}_0(\mathbf{X})$, hence (i_1,j_1) and (i_2,j_2) are not contained in an alternating four-cycle in $\mathcal{B}(\mathbf{X})$. Therefore, alternating C_4-free matchings in $\mathcal{B}(\mathbf{X})$ are exactly the isolated sets in $\mathbf{X}$ and $\mathfrak{i}(\mathbf{X})$ is equal to the cardinality of a maximum alternating C_4-free matching in $\mathcal{B}(\mathbf{X})$.

Example 1.5.1. *Let* $\mathbf{X}$ *be the binary matrix shown in Display* (1.5.1) *below. The bipartite representation* $\mathcal{B}(\mathbf{X})$ *is shown in Figure 1.1a. In the middle of Display* (1.5.1) *we highlight a maximal rectangle indexed by* $\{1,2\} \times \{1,3\}$, *whose biclique equivalent in* $\mathcal{B}(\mathbf{X})$ *is highlighted in Figure 1.1b. On the right side of Display* (1.5.1), *we highlight a maximum isolated set of* $\mathbf{X}$ *and in Figure 1.1c the corresponding maximum* C_4*-free matching of* $\mathcal{B}(\mathbf{X})$ *is highlighted.*

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & \\ 1 & & 1 & 1 \\ 1 & 1 & & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 1 & 1 & \\ 1 & & 1 & 1 \\ 1 & 1 & & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 1 & 1 & \\ 1 & & 1 & 1 \\ 1 & 1 & & 1 \end{bmatrix}. \qquad (1.5.1)$$



(a) $\mathcal{B}(\mathbf{X})$ (b) A maximal biclique (c) A maximum alternating C_4-free matching

Figure 1.1: Bipartite representation of a binary matrix

While we will not use the following perspective in the rest of the thesis, for completeness we mention that $\mathcal{B}(\mathbf{X})$ can be defined for generalised binary matrices as follows. If $\mathbf{X}$ is a generalised binary matrix, then let $\mathcal{B}(\mathbf{X})$ be a generalisation of bipartite graphs in which the adjacency between some vertices is undecided. We may visualise these *generalised graphs* as graphs with a new type of 'undecided' edges. These undecided edges then are used to represent entries $(i,j) \in \mathrm{supp}_?(\mathbf{X})$. The edge set of a biclique of a generalised graph is then a subset of the standard and undecided edges, while the equivalent of an isolated set is a subset of the standard edges that cannot be covered by a common biclique. Such generalised graphs are sometimes called *trigraphs* [18].

k**-BMF.** We may also define rank-k binary matrix factorisation in the bipartite setting on a different bipartite graph. In this case, it is more useful to associate $\mathbf{X} \in \{0, 1, ?\}^{m\times n}$ with an edge weighted complete bipartite graph $(K_{m,n}, \boldsymbol{\mathcal{W}})$ with weight matrix $\boldsymbol{\mathcal{W}} \in \{0, \pm 1\}^{m\times n}$ as defined in Equation (1.4.3), so that each edge (i, j) of $K_{m,n}$ has weight $\mathcal{W}_{i,j}$. An optimal rank-k factorisation of $\mathbf{X}$ corresponds to a maximum weight covering of $(K_{m,n}, \boldsymbol{\mathcal{W}})$ using at most k bicliques and $\zeta(\mathbf{X}, k)$ equals the weight of the maximum weight covering subtracted from $|\operatorname{supp}_1(\mathbf{X})|$ according to Equation (1.4.1). Note that in the bipartite setting, 1-BMF is simply the weighted version of the maximum edge biclique problem called the *maximum weight edge biclique problem*, in which one needs to compute a maximum weight edge biclique of the weighted graph $(K_{m,n}, \boldsymbol{\mathcal{W}})$.

Example 1.5.2. *Let* $\mathbf{X}$ *be the binary matrix as shown below, and let* $\boldsymbol{\mathcal{W}}$ *be its weight matrix,*

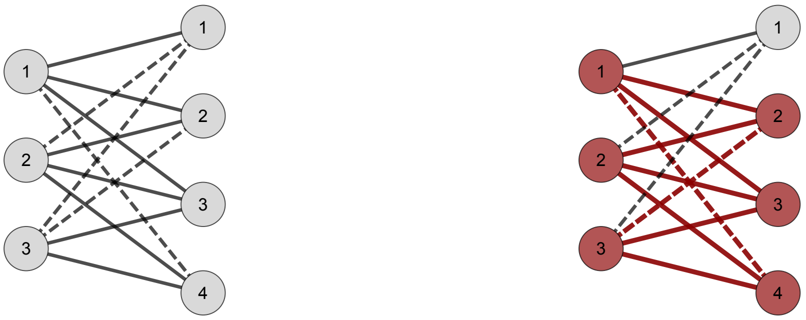
$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & \\ & 1 & 1 & 1 \\ & & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{\mathcal{W}} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}.$$

The weighted graph $(K_{3,4}, \boldsymbol{\mathcal{W}})$ *is shown in Figure 1.2a where dashed edges have weight* -1 *and solid edges have weight* $+1$*. The optimal rank-*1 *factorisation* $\boldsymbol{a}\boldsymbol{b}^\top$ *of* $\mathbf{X}$ *is*

$$\boldsymbol{a}\boldsymbol{b}^\top = \begin{bmatrix} & 1 & 1 & 1 \\ & 1 & 1 & 1 \\ & 1 & 1 & 1 \end{bmatrix},$$

and the corresponding maximum weight edge biclique is highlighted in red in Figure 1.2b. The optimal factorisation error, which equals the number of 1*s of* $\mathbf{X}$ *not covered, and the number of* 0*s of* $\mathbf{X}$ *erroneously covered, can also be written using Equation* (1.4.2)*, as the weight of the maximum weight edge biclique of* $(K_{3,4}, \boldsymbol{\mathcal{W}})$ *subtracted from* $|\operatorname{supp}_1(\mathbf{X})|$*,*

$$\zeta(\mathbf{X}, 1) = 3 = |\operatorname{supp}_1(\mathbf{X})| - \boldsymbol{a}^\top \boldsymbol{\mathcal{W}} \boldsymbol{b} = 8 - (7 - 2).$$



(a) The weighted graph $(K_{3,4}, \mathcal{W})$ (b) Maximum weight biclique of $(K_{3,4}, \mathcal{W})$

Figure 1.2: The bipartite representation of k-BMF

1.6 Complexity

Many complexity results related to the computation of the Boolean rank, isolation number and k-BMF of binary matrices appear in the literature in the bipartite graph setting.

Minimum biclique cover and Boolean rank. The minimum Biclique Cover problem on bipartite graphs (problem BC) is first analysed by Orlin in 1977, and is proved to be NP-hard [91, Theorem 8.1] by reducing the minimum Clique Partition problem (CP) to it. BC is also listed in [38] as Problem GT18. BC remains NP-hard on chordal bipartite graphs [88, Theorem 6]. Furthermore, the closely related problem of partitioning the edge set of a bipartite graph into a minimum number of bicliques is NP-hard as well [53].

In 1990, Simon [102, Corollary 5.2] shows by presenting a continuous reduction between CP and BC that the two problems are approximation equivalent. Minimum Clique Partition is equivalent to minimum Graph Colouring (GC) as $\theta(G) = \chi(\overline{G})$, so the same approximation hardness results hold for these two problems. There have been many inapproximability results proved for GC since the 90s. In 2007, Zuckerman [109, Theorem 1.2] showed that for all $\epsilon > 0$, approximating the chromatic number of any graph $G = (V, E)$ within a factor $\mathcal{O}(|V|^{1-\epsilon})$ is NP-hard. Building on this approximation hardness of GC and improving the reduction between BC and CP, several authors prove hardness results for BC. Gruber et al. [48] show by combining the continuous reduction of Simon with the $\mathcal{O}(|V|^{1-\epsilon})$ inapproximability of GC that for all $\epsilon > 0$ it is NP-hard to approximate BC within a factor $\mathcal{O}(|V|^{\frac{1}{5}-\epsilon})$. In the same work, Gruber et al. [48] also give an improved reduction between CP and BC, which then shows that the biclique cover number of any bipartite graph $G = (V, E)$ cannot be approximated within $\mathcal{O}(|V|^{\frac{1}{3}-\epsilon})$ and $\mathcal{O}(|E|^{\frac{1}{5}-\epsilon})$ for any $\epsilon > 0$. Finally, in 2015 Chalermsook et al. [15] proves that the biclique cover number of any bipartite graph $G = (V, E)$ is NP-hard to approximate within a factor $\mathcal{O}(|V|^{1-\epsilon})$ and $\mathcal{O}(|E|^{\frac{1}{2}-\epsilon})$ for any $\epsilon > 0$. Furthermore, Chalermsook et al. [15, Theorem 2.] shows that these hardness results are nearly optimal by presenting a factor $\mathcal{O}(\frac{|V_1|}{\sqrt{\log |V_1|}})$ and a factor $\mathcal{O}(\frac{|E|(\log \log |E|)^2}{\log^3 |E|})$-approximation algorithms for the biclique cover problem for any bipartite graph $G = (V_1, V_2, E)$ with $|V_1| \leq |V_2|$ (the second of which relies on the best known approximation algorithm for GC by Halldórsson [51]). In 2016, Chandran et al. [16, Theorem 21] give an improved very simple approximation algorithm for BC with ratio $\frac{|V_1|}{\log |V_1|}$. We present this in matrix form. Let $\mathbf{X} \in \{0, 1\}^{m \times n}$ and $2 \leq m \leq n$. Assume that $\mathbf{X}$ has no repeated rows so the bound given in Equation (1.1.7) holds for $\mathbf{X}$ and we have,

$$\log m \leq \mathfrak{br}(\mathbf{X}).$$

Taking the rows of $\mathbf{X}$ gives a feasible factorisation of size m. Then the ratio between the cardinality of this feasible factorisation and an optimal factorisation is

$$\frac{m}{\mathfrak{br}(\mathbf{X})} \leq \frac{m}{\log m}.$$

Since clearly row and column duplicates can be eliminated and replaced into the factorisation in polynomial time, the above algorithm is a factor $\frac{m}{\log m}$-approximation for BC.

BC has also been analysed from a parameterised complexity perspective. It is shown that deciding whether the biclique cover number of a bipartite graph $G = (V, E)$ is less than or equal to k is fixed parameter tractable in k [34] and can be solved in time $\mathcal{O}(f(k) + |V|^3)$ where $f(k) = 2^{k2^{k-1}+3k}$ [90, Theorem 4]. On the other hand, Chandran et al. [16, Corollary 7] show that there is no parameterised algorithm for BC with running time $2^{2^{o(k)}}(mn)^{\mathcal{O}(1)}$ [1] unless the Exponential Time Hypothesis (ETH) is false (ETH is the hypothesis that 3-SAT cannot be solved in time $2^{o(n)}$.)

Maximum alternating C_4-free matching and isolation number. Let us turn to the complexity of the maximum isolated set which in the bipartite setting is the Alternating C_4-free Matching problem on bipartite graphs (AC4M). AC4M is first examined by Pulleyblank [95] in 1982 as a special case of alternating cycle-free matchings,

[1] $f(x) \in o(g(x))$ means that for *every* $\epsilon > 0$, there exists a constant x_0 such that $|f(x)| < \epsilon g(x)$ holds for all $x > x_0$. While $f(x) \in \mathcal{O}(g(x))$ means that there exist constants $k > 0$ and x_0 such that $|f(x)| < kg(x)$ holds for all $x > x_0$. For instance, $x^2 \in \mathcal{O}(x^2)$ but $x^2 \notin o(x^2)$ and $x^2 \in o(x^3)$.

and is shown to be NP-hard by reducing 3D-matching to it. (The 3D-matching problem asks to find a maximum cardinality subset M of a hypergraph $T \subset [m] \times [n] \times [p]$ in which any distinct $(i_1, j_1, t_1), (i_2, j_2, t_2) \in M$ satisfies $i_1 \neq i_2$, $j_1 \neq j_2$ and $t_1 \neq t_2$.) AC4M remains NP-hard on chordal bipartite graphs [87, Section 6]. Unfortunately, we are not aware of any works on further hardness results or approximation algorithms for AC4M.

There are several classes of binary matrices for which the isolation number and Boolean rank can be solved in polynomial time. Interestingly, all these matrix classes for which polynomial time algorithms are known are firm matrices. These classes are matrices with the consecutive 1s property [50], linear matrices, biadjacency matrices of distance hereditary bipartite graphs [77], and biadjacency matrices of domino-free bipartite graphs [2]. We detail these classes more in depth in Section 3.3.

Maximum edge biclique and maximum rectangle. Since bicliques of bipartite graphs are in direct correspondence with rectangles of binary matrices, the maximum edge biclique problem on bipartite graphs is equivalent to the maximum rectangle problem of binary matrices. The decision version of the Maximum Edge Biclique problem (MEB) with input bipartite graph $G = ([m], [n], E)$, $m \leq n$ and positive integer k, asks if G contains a biclique with at least k edges. This problem is proved to be NP-complete in 2003 by Peeters [93] by reducing the maximum clique problem to it. For chordal bipartite graphs MEB is polynomial time solvable [41].

There are several inapproximability results for MEB but all of them are conditional on some special complexity assumptions that are stronger than the standard assumption P$\neq$NP, see for instance [32, Theorem 3],[1, Theorem 1.4]. The latest of these results proves that MEB is hard to approximate within $\mathcal{O}(m^{1-\epsilon})$ for any $\epsilon > 0$ given that special complexity assumptions hold [81].

Maximum weight edge biclique. Several weighted versions of MEB are analysed too. Let us denote the Maximum Weight Edge Biclique problem (MWEB) with edge weights from a set $\mathcal{Q}$ by $\mathcal{Q}$-MWEB. Note that if the input bipartite graph $G = ([m], [n], E)$ for $\mathcal{Q}$-MWEB is not a complete bipartite graph then we can create an equivalent instance $\{-M\} \cup \mathcal{Q}$-MWEB on $K_{m,n}$ where edges in E have their original weight and edges $(i, j) \notin E$ have weight $-M$, where M is some constant that is larger than the total weight of all positively weighted edges in E. In addition, if the input graph for $\mathcal{Q}$-MWEB is $K_{m,n}$ then $\mathcal{Q}$ must contain both negative and positive numbers, as otherwise the problem is trivial.

Dawande et al. [27] looked at $\{0,1\}$-MWEB on general bipartite graphs in 1996 and proved it to be NP-hard. Their reduction only holds if 0 weights are present so this early result did not imply the hardness of the cardinality version MEB (which is just $\{1\}$-MWEB on general bipartite graphs and is proved to be NP-hard by Peeters [93] later in 2003 as mentioned above). Some inapproximability results for $\{0,1\}$-MWEB on general bipartite graphs are also shown based on the relation of $\{0,1\}$-MWEB to the minimum Biclique Cover problem. Simon [102, Section 6.] argues that if $\{0,1\}$-MWEB could be solved exactly then it could be used in a master-slave algorithm to give an $\mathcal{O}(\ln(|E|))$ approximation of BC on a bipartite graph G with edge set E. Chalermsook [15, Corollary 2.] shows that combining this approach with the inapproximability results of BC, $\{0,1\}$-MWEB on a bipartite graph $G = ([m],[n],E)$, is hard to approximate within $\mathcal{O}(m^{1-\epsilon})$ and $\mathcal{O}(|E|^{\frac{1}{2}-\epsilon})$ for all $\epsilon > 0$ unless P=NP.

$\{-M,1\}$-MWEB on $K_{m,n}$ for any sufficiently large M is NP-hard as it can encode MEB. Tan [103, Lemma 4, Theorem 1.] shows that $\{-1,0,1\}$-MWEB on $K_{m,n}$ is NP-hard and for every $\epsilon > 0$, it cannot be approximated within $\mathcal{O}((m+n)^{1-\epsilon})$ unless P=NP. Gillis et al. [39, Corollary 4] prove that the even more restricted $\{-1,1\}$-MWEB is NP-hard as well.

1-BMF. Recall that in the previous sections, we argued that rank-1 binary matrix factorisation on a generalised binary matrix $\mathbf{X} \in \{0,1,?\}^{m\times n}$ is equivalent to a Maximum Weight Edge Biclique problem on $K_{m,n}$ with edge weights $\mathcal{W}_{i,j} \in \{0,\pm 1\}$ as defined in Equation (1.4.3). Therefore, as $\{-1,0,1\}$-MWEB on $K_{m,n}$ is NP-hard, rank-1 BMF of binary matrices with missing entries is NP-hard too. Furthermore, by Gillis et al.'s result [39, Corollary 4], rank-1 BMF of standard binary matrices is NP-hard as well. On the other hand, there is a simple 2-approximation algorithm [101] for 1-BMF that we show in Section 8.3.1.

k-BMF. The hardness of 1-BMF implies that k-BMF is NP-hard too by restriction. In addition, one can also see the NP-hardness of k-BMF as k-BMF is harder than computing the Boolean rank as by solving logarithmically many k-BMF problems one could compute the Boolean rank, but not the other way round. Furthermore, as BC is fixed parameter tractable in k, deciding whether the Boolean rank is less than or equal to k can be solved in polynomial time for any fixed k; while k-BMF is NP-hard already for $k = 1$.

Regarding approximation algorithms for k-BMF, [85] observed the following argument. If there is an α-approximation algorithm for k-BMF of an $m \times n$ matrix $\mathbf{X}$

then it produces a feasible rank-k factorisation $\mathbf{A}' \circ \mathbf{B}'$ which satisfies

$$\|\mathbf{X} - \mathbf{A}' \circ \mathbf{B}'\|_F^2 \leq \alpha \cdot \min_{\mathbf{A} \in \{0,1\}^{m \times k}, \mathbf{B} \in \{0,1\}^{k \times n}} \|\mathbf{X} - \mathbf{A} \circ \mathbf{B}\|_F^2.$$

Thus any approximation algorithm for k-BMF must be able to distinguish between $\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{A} \circ \mathbf{B}\|_F^2 = 0$ and $\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{A} \circ \mathbf{B}\|_F^2 > 0$. The parameterised complexity results on BC [16] tell us that this is not plausible in time $2^{2^{o(k)}}(mn)^{\mathcal{O}(1)}$ unless ETH is false.

Part I

Exact Binary Matrix Factorisation

Chapter 2

Introduction

In this part of the thesis, we look at the Boolean rank and isolation number from a combinatorial perspective as a covering and packing problem pair. For any binary matrix $\mathbf{X}$, an isolated set S and a rectangle $I \times J$ can intersect at at most one element $|S \cap (I \times J)| \leq 1$, which also shows that $\mathfrak{i}(\mathbf{X}) \leq \mathfrak{br}(\mathbf{X})$. We are interested in matrices where the weak duality between $\mathfrak{br}(\mathbf{X})$ and $\mathfrak{i}(\mathbf{X})$ becomes strong and this also holds for their submatrices. A binary matrix $\mathbf{X}$ is said to be *firm* if $\mathfrak{i}(\mathbf{X}) = \mathfrak{br}(\mathbf{X})$ and $\mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}')$ holds for all submatrices $\mathbf{X}'$ of $\mathbf{X}$. The concept of firmness along with many results that motivate and form the basis of our work are introduced in a 1990 paper [77] of Anna Lubiw. A key tool that Lubiw introduces and we make extensive use of, is to look at the problem through the rectangle cover graph of $\mathbf{X}$. The *rectangle cover graph* $\mathcal{G}(\mathbf{X})$ of a binary matrix $\mathbf{X}$ (called the *1's graph* in Lubiw's words), has a vertex (i, j) for each 1 at $(i, j) \in \operatorname{supp}_1(\mathbf{X})$ and an edge between two vertices if and only if the corresponding 1s in $\mathbf{X}$ can be covered by a common rectangle. Isolated sets of $\mathbf{X}$ then correspond to independent sets of $\mathcal{G}(\mathbf{X})$, and maximal rectangles of $\mathbf{X}$ to maximal cliques of $\mathcal{G}(\mathbf{X})$ [77]. By this, $\mathfrak{i}(\mathbf{X})$ and $\mathfrak{br}(\mathbf{X})$ translate to the independence and clique cover number of $\mathcal{G}(\mathbf{X})$, respectively, and one can explore the parallels between firmness of $\mathbf{X}$ and perfection of $\mathcal{G}(\mathbf{X})$: A binary matrix $\mathbf{X}$ is said to be *superfirm* if $\mathcal{G}(\mathbf{X})$ is a perfect graph [77]. It turns out that perfection of the rectangle cover graph is a stronger requirement than firmness and superfirm matrices form a strict subset of firm matrices.

To get a better understanding of firmness and superfirmness, one might try to list the minimal violators of these properties. The investigation of minimal violators is a common approach in combinatorics and has been applied to the study of perfect graphs via minimally imperfect graphs or ideal matrices via minimally non-ideal matrices [79, 105]. Motivated by this, we start the explicit study of firm and superfirm matrices through forbidden submatrices. Forbidding a submatrix $\mathbf{X}$ means that we look at the class of matrices which cannot have **X** as a submatrix in any row or column order. We say that a binary matrix **X** is *minimally non-firm* if **X** is not firm but all of its *proper* submatrices are. Analogously, a binary matrix **X** is said to be *minimally non-superfirm* if **X** is not superfirm but all of its *proper* submatrices are. With these definitions, we can then say that a binary matrix is firm if and only if it does not have any minimally non-firm submatrix, and it is superfirm if and only if it does not have any minimally non-superfirm submatrix.

The holy grail of this direction would be to be able to characterise firm matrices by a complete set of minimally non-firm matrices and superfirm matrices by a complete set of minimally non-superfirm matrices. We are very far from this. But to the best of our knowledge, minimally non-firm and minimally non-superfirm matrices have not been explicitly studied before and we present the first infinite classes of minimally non-firm matrices.

Our contribution and Organisation of Part I. Let us give a detailed summary of our contributions and the organisation of Part I.

In the remaining sections of this chapter, we walk through the history of firm matrices which originates in the study of rectilinear polygons. Then we present some problems related to firmness of binary matrices.

In Chapter 3, first we give an in depth summary of Lubiw's work [77]. In particular, we go through the definition of rectangle cover graphs, superfirmness and firmness in detail and illustrate these concepts through examples. Then we continue Lubiw's approach in exploring the parallels between perfect graphs and firm matrices. In Section 3.3, we give an in-depth review of the techniques used to prove the so far known classes of firm matrices.

In Chapter 4, we explore how minimally imperfect subgraphs can appear in rectangle cover graphs. We prove that odd antiholes cannot appear in rectangle cover graphs without odd holes being present. This shows that the property of superfirmness is equal to not having any odd holes in the rectangle cover graph and forbidding odd antiholes is unnecessary. Then we characterise the submatrices which are necessary and sufficient for the appearance of 5-holes in rectangle cover graphs. Along these lines, we also prove that P_5-free rectangle cover graphs are perfect and present several minimally non-superfirm matrices.

In the second part of Chapter 4, we define simplicial 1s and a procedure for their removal which leads to generalised binary matrices. Then, as one of our main contributions, we introduce a matrix operation called 'stretching'. We show that under some conditions stretching preserves firmness and superfirmness.

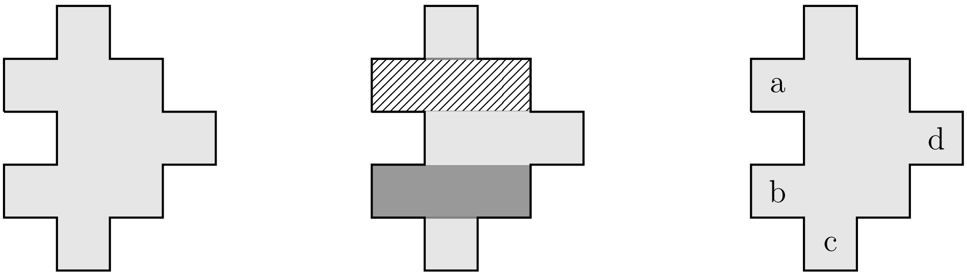
In Chapter 5, we use the stretching operation to derive a theorem which gives a general recipe on how to create minimally non-firm matrices from matrices that have odd holes in their rectangle cover graphs and satisfy certain conditions. Then we apply this theorem to obtain several infinite families of minimally non-firm matrices. While there are many open questions and holes left on the way, we hope that our work may motivate someone else in the future to pick up the study of minimally non-firm matrices.

Finally, in Chapter 6 we conclude and state several open questions.

2.1 Rectilinear polygons

The study of firm matrices originates in the study of rectilinear polygons or sometimes called polymonios. A *rectilinear polygon* is a polygon in the plane with horizontal and vertical sides which has a single continuous boundary, i.e. no holes are allowed. See Figure 2.1a for an example. A *rectangle* [77] of a rectilinear polygon has horizontal and vertical sides and it is continuous, so that it is fully contained inside the polygon. To distinguish from the rectangles of binary matrices, we will refer to rectangles of rectilinear polygons as *continuous rectangles.* Note that a continuous rectangle of a rectilinear polygon P needs to correspond to a connected region in P, while rectangles of binary matrices need not be defined by contiguous sets of row and column indices. A maximal continuous rectangle is one that is not contained in another continuous rectangle. Two maximal continuous rectangles are indicated in Figure 2.1b. An *antirectangle* [77] of a rectilinear polygon P is a set of points in P no two of which can be covered by a common continuous rectangle. An antirectangle of cardinality four indicated by letters 'a,b,c,d' is shown in Figure 2.1c. Observe that an antirectangle can have elements on a straight line ('a' and 'b'), and this is because only *continuous* rectangles are considered for rectilinear polygons.

Chvátal once conjectured that for all rectilinear polygons P the cardinality of a maximum antirectangle and the minimum number of continuous rectangles needed to cover P is equal (mentioned in [14] in 1981). This conjecture however turned out to be false by Chung providing a counterexample that is shown on the left side of Figure 2.2 (first presented in [14]). Observe that Chung's polygon has five points (which are indicated by letters 'a,b,c,d,e' on the right side of Figure 2.2) each of which can only be covered by one unique maximal continuous rectangle (shaded in the figure).



(a) A rectilinear polygon (b) Two maximal continuous rectangles (c) A maximum antirectangle

Figure 2.1: The rectangle cover problem on rectilinear polygons

Thus these continuous rectangles can be assumed to belong to a minimum cover. The region that is left uncovered has five points (indicated by red vertices) which form a cycle of length 5 in the graph where points are adjacent if they can be covered by a common continuous rectangle. Hence these 5 points need at least three continuous rectangles to be covered. On the other hand, the uncovered region contains only antirectangles of size two. Therefore Chung's polygon has a maximum antirectangle of size 7, but its minimum cover is of size 8.

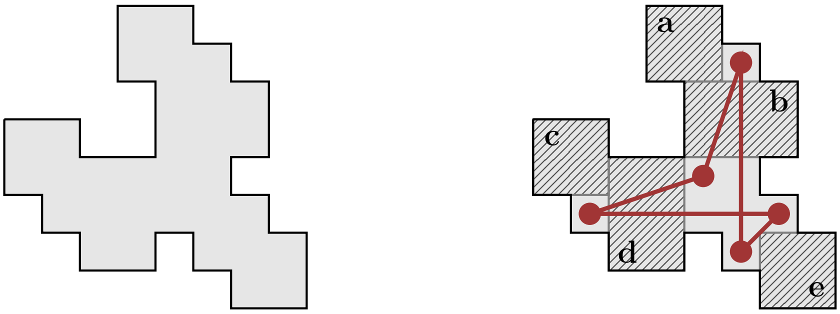


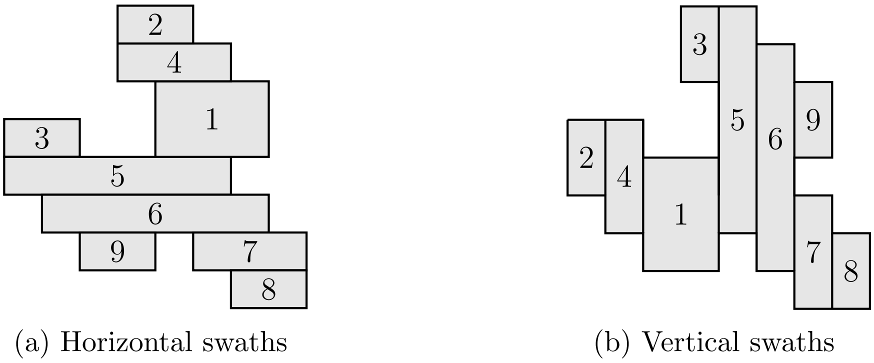
Figure 2.2: Chung's polygon

The discovery of Chung's polygon motivated the adjustment of Chvátal's conjecture. A rectilinear polygon P is said to be x*-convex* (y*-convex*) if every horizontal (vertical) line segment joining two points inside P is contained inside the polygon. For instance the polygon in Figure 2.1a is x-convex but not y-convex, and Chung's polygon is neither x- nor y-convex. In 1981, Chaiken et al. [14] proved that Chvátal's conjecture when restricted to rectilinear polygons that are both x- and y-convex (x, y*-convex*) is true. Observe that this also gives the first class of firm matrices as a binary matrix that has a 1 for every square unit of an x, y-convex rectilinear polygon has only continuous maximal rectangles (so every maximal submatrix of all 1s has the row and column index set as a range of integers).

Later on, this minimax result is extended to a more challenging superset of x, y-convex rectilinear polygons. In 1984 Győri [50] proves a deep result which shows that Chvátal's conjecture holds when restricted to rectilinear polygons that are only y-convex (or equivalently, only x-convex). In fact, the result that Győri proves is a more general theorem which shows that for interval matrices the Boolean rank and the isolation number are equal. A binary matrix is an *interval* matrix if its columns can be arranged so that the 1s in every row appear consecutively. These matrices are sometimes also called matrices with the *consecutive* 1*'s property.* As any submatrix of an interval matrix is also interval, Győri's theorem implies that interval matrices are firm. Interestingly, there are some interval matrices which are not superfirm, so their rectangle cover graph is not perfect. This shows that interval matrices are not a subclass of superfirm matrices. We thoroughly explain this with examples in the next chapter. Following Győri's theorem, a polynomial time algorithm is presented for the factorisation of interval matrices by Franzblau et al. [37] which we will detail in Section 3.3.4.

Rectilinear polygon covering motivated the discovery of the first classes of firm matrices, but it was not until Lubiw's seminal paper [77], where a transformation of rectilinear polygons into binary matrices is given, showing that rectilinear polygon covering is just a special case of our problem on binary matrices. In turn, Lubiw mentions that her idea for this transformation is inspired by the transformation that Frank suggested to Győri and Győri used in [50] to prove his theorem.

Next we illustrate Lubiw's transformation of rectilinear polygons into binary matrices on Chung's polygon. A *horizontal (vertical) swath* of a rectilinear polygon P is a maximum horizontal-length (vertical-length) continuous rectangle whose vertical (horizontal) borders are the borders of P. The horizontal and vertical swaths of Chung's polygon are indicated in Figure 2.3. The *swath matrix* of a rectilinear poly-



(a) Horizontal swaths (b) Vertical swaths

Figure 2.3: Swaths of Chung's polygon

gon P is a binary matrix which has a row for each horizontal swath of P, a column for each vertical swath of P and a 1 in a position if and only if the corresponding horizontal and verticals swaths of P intersect. The swath matrix of Chung's polygon is below,

$$\begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & & & & & 1 & 1 & & & 1 \\
2 & & & 1 & & 1 & & & & \\
3 & & 1 & & 1 & & & & & \\
4 & & & 1 & & 1 & 1 & & & \\
5 & 1 & 1 & & 1 & 1 & 1 & & & \\
6 & 1 & & & 1 & 1 & 1 & 1 & & \\
7 & & & & & & 1 & 1 & 1 & \\
8 & & & & & & & 1 & 1 & \\
9 & 1 & & & & & & & &
\end{array}. \tag{2.1.1}$$

Lubiw proves that if P is a rectilinear polygon and $\mathbf{X}$ is its swath matrix then maximal continuous rectangles of P correspond exactly to the maximal rectangles of $\mathbf{X}$ [77, Proposition 2.1] and antirectangles of P correspond exactly to the isolated sets of $\mathbf{X}$. For instance, the maximal continuous rectangle of Chung's polygon that is shaded and contains point **a** in the right side of Figure 2.2 is intersecting horizontal swaths 2 and 4 and vertical swaths 3 and 5, and the corresponding maximal rectangle of the swath matrix in Display 2.1.1 is indexed by rows $2, 4$ and columns $3, 5$. Similarly, if A is a set of points that forms an antirectangle in a rectilinear polygon P, then every point in A is at the intersection of a distinct horizontal and vertical swath of P, which give the indices of the corresponding 1s in the swath matrix of P that form the corresponding isolated set.

Since one may assume that an optimal cover of P uses only maximal continuous rectangles and an optimal cover of $\mathbf{X}$ also uses only maximal rectangles, the rectilinear polygon cover problem is just a special case of the rectangle cover problem on binary matrices. As an example one can verify that the swath matrix of Chung's polygon (given in Equation (2.1.1)) has Boolean rank 8 and isolation number 7, hence it is a non-firm matrix. In fact, dropping the first and last rows and columns of this matrix we get a *minimally non-firm* matrix as shown below,

$$\begin{array}{c|ccccccc} & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 2 & & 1 & & 1 & & & \\ 3 & 1 & & 1 & & & & \\ 4 & & 1 & & 1 & 1 & & \\ 5 & 1 & & 1 & 1 & 1 & & \\ 6 & & & 1 & 1 & 1 & 1 & \\ 7 & & & & & 1 & 1 & 1 \\ 8 & & & & & & 1 & 1 \end{array} . \qquad (2.1.2)$$

A y-convex rectilinear polygon P has an interval swath matrix $\mathbf{X}$ and this is why firmness of interval matrices implies Chvátal's conjecture on y-convex matrices. To see this, observe that since P is y-convex, for each vertical line segment there is exactly one vertical swath of P, so we may assume that vertical swaths of P are numbered in increasing order from left to right. The 1s in each row of $\mathbf{X}$ correspond to the intersection of a horizontal swath with some consecutively numbered vertical swaths, therefore for each row i if $x_{i,j_1} = 1$ and $x_{i,j_2} = 1$ then $x_{i,j} = 1$ for all $j_1 < j < j_2$. So $\mathbf{X}$ is an interval matrix.

Of course, not all swaths matrices are interval, but Lubiw shows that they are totally balanced [77, Theorem 2.2]. For $n \geq 3$, the $n \times n$ *cycle matrix* $\mathbf{C}_n \in \{0,1\}^{n \times n}$ contains exactly two 1s in every row and in every column and no proper submatrix of $\mathbf{C}_n$ has this property. We adopt the following ordering of $\mathbf{C}_n$ as below,

$$\mathbf{C}_n = \begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{bmatrix}.$$

A binary matrix is said to be *totally balanced* if it has no $\mathbf{C}_n$ submatrix for any $n \geq 3$ [41, Chapter 12.4]. In 1988, it is proved that covering rectilinear polygons by a minimum number of continuous rectangles is NP-hard [24]. By Lubiw's transformation, this in turn implies that covering totally balanced matrices by a minimum number of rectangles is also NP-hard.

2.2 Firmness

The explicit study of firm matrices comes from the 1990 seminal paper of Lubiw [77] in which she also proves that rectilinear polygon covering is just a special case of our problem. In this same paper, she defines firmness, rectangle cover graphs and proves three classes of matrices to be superfirm. The first class is the class of matrices which have no $\mathbf{J}_2$ submatrix (2×2 submatrix of all 1s). $\mathbf{J}_2$-free matrices are usually called *linear* matrices [22]. Linear matrices will be further discussed in Section 3.3.1.

Lubiw introduces a superfirmness preserving operation which we call Lubiw-sum or *L-sum* for short, and mentions that this operation comes from the more general method of split decomposition on bipartite graphs which is due to Cunningham [25]. Briefly, in the setting of binary matrices, the L-sum takes two binary matrices as input and creates a new binary matrix in which the input matrices are joined on a common rectangle which is created from a specified row and column of the input matrices. Using the polynomial time algorithm of Cunningham to compute a split decomposition, Lubiw shows that the Boolean rank and isolation number can be computed in polynomial time for any matrix that can be obtained as a series of L-sums starting from matrices for which there is a polynomial time algorithm to compute the Boolean rank and isolation number.

The second class of matrices proved by Lubiw to be superfirm is the class of matrices that can be obtained via a sequence of L-sum operations starting on binary row and column vectors. Then Lubiw extends this class to show that a third class of matrices, matrices that can be obtained via a sequence of L-sum operations starting on linear matrices, is also superfirm. In addition, Lubiw also proves a forbidden submatrix characterisation for this class of matrices. The L-sum operation and these two classes of superfirm matrices will be discussed more in detail in Section 3.3.2.

In addition, Lubiw presents two non-firm matrices in [77, Figure 1.1.] to illustrate that the Boolean rank does not always equal the isolation number. Both of these matrices turn out to be minimally non-firm, although Lubiw does not mention this. One of these matrices comes from the swath matrix of Chung's polygon that we show in Equation (2.1.2). Several other results are treated in [77], by which Lubiw lays down the foundations for forbidden submatrix characterisation of firm and superfirm matrices. We make an immense use of the techniques Lubiw developed.

$\mathbf{D}_3$-free binary matrices are the latest and largest class of matrices proved to be firm by Amilhastre et al. in 1998 [2]. Their proof is given in the bipartite setting and shows that for domino-free bipartite graphs a minimum biclique cover can be computed in polynomial time. The *domino graph* is a cycle on six vertices with exactly one chord as shown in Figure 2.4. Let us denote the domino graph by $\mathcal{B}(\mathbf{D}_3)$

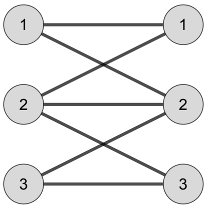


Figure 2.4: The domino graph, $\mathcal{B}(\mathbf{D}_3)$

through its biadjacency matrix, which is given by

$$\mathbf{D}_3 = \begin{bmatrix} 1 & 1 & \\ 1 & 1 & 1 \\ & 1 & 1 \end{bmatrix}. \tag{2.2.1}$$

In matrix terms, Amilhastre et al. show that a minimum rectangle cover of $\mathbf{D}_3$-free matrices can be computed in polynomial time. Interestingly, the isolation number of $\mathbf{D}_3$-free matrices is not explored by Amilhastre et al. However, their results with a slight extension implies that $\mathbf{D}_3$-free matrices are firm. In fact, one can find $\mathbf{D}_3$-free matrices that are not superfirm. Hence, $\mathbf{D}_3$-free matrices give the second class of firm but not necessarily superfirm matrices, the first one being interval matrices. In Section 3.3.3 we will detail the methods of Amilhastre et al. and the firmness of $\mathbf{D}_3$-free matrices.

2.2.1 Firmness in later works

While Lubiw's 1990 paper seems to be the earliest to introduce the concept of firm matrices, several other later works reintroduce the definition of firm matrices under new names and seem to be unaware of the work of Lubiw.

Muller is one of these authors investigating concepts related to firmness in two papers [87, 88]. In [87], Muller proves that the maximum alternating C_4-free matching problem remains NP-hard when restricted to chordal bipartite graphs [87, Sect. 4]. A bipartite graph is said to be *chordal bipartite* if every cycle of length at least 6 has a chord [41, Chapter 12.4]. Recall that $\mathbf{C}_n$ is the $n \times n$ cycle matrix for $n \geq 3$ and a matrix is totally balanced if it does not have any $\mathbf{C}_n$ submatrices. In the bipartite setting, $\mathbf{C}_n$ are the biadjacency matrices of chordless cycles of length $2n$. Hence, the bipartite representation of totally balanced matrices are exactly the chordal bipartite graphs. Consequently, in the binary matrix setting Muller's result shows that computing a maximum isolated set of totally balanced matrices is NP-hard.

In [87], Muller also defines rectangle cover graphs under the name *dependence graphs* and investigates how chordless cycles can appear in them. In [88], Muller then looks at minimum biclique covers and redefines firmness in the bipartite setting under the name of *edge-perfection* and superfirmness under the name d-perfection without being aware of Lubiw's previous results. Muller also gives a new proof that the minimum biclique cover problem is NP-hard on chordal bipartite graphs[88, Sect. 8].

Another author, Phelps, who also seems to be unaware of Lubiw's work, treats firmness under the name *factor perfection* and superfirmness under the name *graphical factor perfection* in his 1996 PhD thesis [94]. While this thesis contains several interesting results (we adopted the name *rectangle cover graph* for $\mathcal{G}(\mathbf{X})$ from this thesis), one of the main results of Phelps is flawed, which incorrectly states that totally balanced matrices are firm.

Finally in 2003, Dawande [26] investigates superfirmness in the bipartite setting under the name *cross perfection* from a polyhedral perspective. He defines a bipartite graph $G = ([n], [m], E)$ to be *cross-perfect* if for every cost function $\boldsymbol{c} \in \{0, 1\}^{|E|}$,

$$\max\{\boldsymbol{c}^\top \boldsymbol{y} : \boldsymbol{\mathcal{K}} \boldsymbol{y} \leq \mathbf{1}, \boldsymbol{y} \in \{0, 1\}^{|E|}\} = \min\{\mathbf{1}^\top \boldsymbol{q} : \boldsymbol{\mathcal{K}}^\top \boldsymbol{q} \geq \boldsymbol{c}, \boldsymbol{q} \in \{0, 1\}^p\},$$

where G has p maximal bicliques and $\boldsymbol{\mathcal{K}}$ is a binary matrix with a row for each maximal biclique of G, column for each edge $(i, j) \in E$ and $\mathcal{K}_{B,(i,j)} = 1$ if edge (i, j) is in biclique B. Dawande proves that $\{\boldsymbol{y} \geq \mathbf{0} : \boldsymbol{\mathcal{K}} \boldsymbol{y} \leq \mathbf{1}\}$ is an integral polytope if and only if G is cross-perfect by using the concept of a *modified line graph* (which is another redefinition of rectangle cover graphs) and derives that cross-perfection is exactly this modified line graph being perfect. Therefore, cross-perfection is precisely superfirmness.

2.3 Related problems

2.3.1 Weakly firm matrices

A binary matrix is called *weakly firm* if its Boolean rank equals its isolation number but this equality may not hold for some submatrices of it. Similarly, to the case of weakly perfect graphs (graph G is *weakly perfect* if $\alpha(G) = \theta(G)$ but this equality may not hold for some induced subgraphs of G), the problem of characterising weakly firm matrices in terms of forbidden submatrices is ill defined. We show this by an example that is inspired by an observation in [41, pg 261]. Let $\mathbf{X}$ be an arbitrary

$m \times n$ binary matrix and let $\mathbf{X}'$ be an $m \times (m+n)$ matrix which consists of two blocks: $\mathbf{X}$ and an identity matrix $\mathbf{I}_m$,

$$\mathbf{X}' = \begin{bmatrix}\mathbf{I}_m & \mathbf{X}\end{bmatrix}.$$

$\mathbf{X}'$ has m rows, so it can be covered by m rectangles and the 1s of block $\mathbf{I}_m$ form an isolated set of size m, hence $\mathbf{X}'$ is weakly firm. This construction shows that starting from an arbitrary binary matrix $\mathbf{X}$, we can always construct a weakly firm matrix which has $\mathbf{X}$ as its submatrix, hence we cannot expect to get a characterisation of weakly firm matrices in terms of forbidden submatrices.

2.3.2 Maximum rectangle and minimum cover by isolated sets.

Two problems related to the minimum rectangle cover and maximum isolated set problems on binary matrices are the problems of finding a maximum rectangle and covering the 1s of the binary matrix by a minimum number of isolated sets. These problems are easily seen to be weak duals of each other and one might wonder if for firm matrices the cardinality of the maximum rectangle equals the size of minimum cover by isolated sets. It turns out that this is not the case.

A first counterexample in which the minimum number of isolated sets needed to cover the 1s is strictly larger than its maximum rectangle is presented by Boucher in [11] in the setting of an x, y-convex rectilinear polygon originally. We present a simplified version of Boucher's counterexample in matrix form here[1]. By duplicating rows and columns of a matrix, the Boolean rank and isolation number do not change. On the other hand, by duplicating rows and columns the maximum rectangle and the minimum cover by isolated sets evidently grow but they do not necessarily grow by the same amount and this is what can be exploited to build a counterexample.

Let $\mathbf{D}_4$ be the 4×4 *row-column interval* (having the consecutive 1s property for both the rows and columns) matrix given below, where empty entries correspond to 0s,

$$\mathbf{D}_4 = \begin{bmatrix} & 1 & 1 & \\ 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 \\ & & 1 & 1 \end{bmatrix}. \tag{2.3.1}$$

Duplicating each row and column of $\mathbf{D}_4$ the number of times the integer row and columns weights as assigned on the left hand side in Display (2.3.2) below, we get a

[1] We thank Prof Colin McDiarmid for suggesting the simplest weight setting that we use to demonstrate Boucher's example in matrix form.

matrix which is still row-column interval and has three maximum rectangles of size $\mathit{36}$ which correspond to the rectangles $\{3,4\} \times \{3,4\}$, $\{3\} \times \{1,2,3,4\}$, $\{1,2,3,4\} \times \{3\}$ in the original matrix $\mathbf{D}_4$,

$$\begin{array}{c|cccc|} & \mathit{2} & \mathit{1} & \mathit{4} & \mathit{2} \\ \mathit{2} & & 1 & 1 & \\ \mathit{1} & 1 & 1 & 1 & \\ \mathit{4} & 1 & 1 & 1 & 1 \\ \mathit{2} & & & 1 & 1 \end{array} \longrightarrow \begin{bmatrix} & \mathit{2} & \mathit{8} & \\ \mathit{2} & \mathit{1} & \mathit{4} & \\ \mathit{8} & \mathit{4} & \mathit{16} & \mathit{8} \\ & & \mathit{8} & \mathit{4} \end{bmatrix}. \tag{2.3.2}$$

Covering by isolated sets can be seen as colouring the 1s in which every isolated set is a unique colour. Each 1 of $\mathbf{D}_4$ is duplicated into a rectangle of the size shown on the right hand size of Display (2.3.2) above, so we may refer to these rectangles by the index of the 1 they were duplicated from (e.g. the rectangle in the right bottom corner is of size $\mathit{4}$ and is referred to by index $(4,4)$).

Let A, B, C and D be a set of distinct $\mathit{36} = \mathit{8} + \mathit{8} + \mathit{4} + \mathit{16}$ colours. Without loss of generality, we can colour $(3,4)$ with $\mathit{8}$ colours in set A, $(4,3)$ with other $\mathit{8}$ colours in set B, $(4,4)$ with $\mathit{4}$ colours in set C and $(3,3)$ with $\mathit{16}$ colours in set D. Then we have so far used $\mathit{36}$ colours. As row 3 is also a maximum rectangle of size $\mathit{36}$, and the uncoloured rectangles at $(3,1)$ and $(3,2)$ cannot get colours A and D, colour them with colours B and C (use a subset B' of size $\mathit{4}$ to colour $(3,2)$). Similarly, column 3 is a maximum rectangle of size $\mathit{36}$, so we can colour its remaining entries with colours A and C. So far we have the colouring,

$$\begin{bmatrix} & \mathit{2} & C \cup A \setminus A' & \\ \mathit{2} & \mathit{1} & A' & \\ C \cup B \setminus B' & B' & D & A \\ & & B & C \end{bmatrix}.$$

At this point however, entry $(2,2)$ is uncoloured and it is in a common rectangle with all the colours we have used so far, so $\mathit{36}$ isolated sets do not suffice to cover all the 1s of the matrix.

On the other hand, by Chaiken et al.'s theorem [14] row-column interval matrices are firm, so $\mathbf{D}_4$ is firm and row-column duplication preserves firmness so the matrix obtained after row-column duplication is firm too.

Therefore, Boucher's example shows an important difference between the two related problem pairs: row and column duplication preserves firmness, but it does not preserve equality between the maximum rectangle and the minimum cover by isolated sets.

2.3.3 Maximum rectangle in totally balanced matrices

The matrix that we considered above is row-column interval, so clearly totally balanced. It is interesting to see that yet there is a polynomial time algorithm to compute the maximum rectangle of totally balanced matrices. This algorithm is heavily dependent on the polynomial time algorithm testing for totally balancedness. Let $\mathbf{\Gamma}$ be the 2×2 binary matrix,

$$\mathbf{\Gamma} := \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

A binary matrix $\mathbf{X}$ has a $\mathbf{\Gamma}$*-free ordering* if its rows and columns can be permuted so that it has no $\mathbf{\Gamma}$-submatrix where the order of the rows and columns of $\mathbf{\Gamma}$ is the same as in the ordering of $\mathbf{X}$. The following theorem leads to an efficient polynomial time recognition algorithm of totally balanced matrices.

Theorem 2.3.1. *[3, 76] A binary matrix* $\mathbf{X}$ *is totally balanced if and only if it has a* $\mathbf{\Gamma}$*-free ordering.*

Lubiw gives an algorithm in [76] which computes a *doubly-lexical ordering* of a binary matrix in polynomial time and shows that a doubly-lexical ordering is $\mathbf{\Gamma}$-free if the matrix is totally balanced.

Most of the results around computing the maximum rectangle in binary matrices are presented in the bipartite setting as the Maximum Edge Biclique problem. So let us go back to the bipartite setting and let $G = (V_1, V_2, E)$ be a bipartite graph. An edge $(x, y) \in E$ of a bipartite graph G is said to be *bisimplicial* if the graph induced by vertices x and y and their neighbours is a biclique [42][41, pg. 256]. Let $\sigma = \{e_1, e_2, \dots, e_{|E|}\}$ be an ordering of the edges of G. σ is said to be a *perfect edge-without-vertex erasing order (pewve)* of G if for each i, e_i is a bisimplicial edge in the graph obtained by deleting edges $\{e_1, \dots, e_{i-1}\}$ [41, pg. 298]. How does a pewve order of G relate to the maximum edge biclique of G? It follows from the definition that a bisimplicial edge can be only contained in exactly one maximal biclique. Therefore, if a pewve order is available for a bipartite graph G, going through and deleting the edges as in the pewve order and recording the maximal biclique corresponding to each edge, we obtain a list of size $|E|$ containing all maximal bicliques of G [56]. Afterwards, by counting the number of edges in each maximal biclique in this list of size $|E|$, it is straightforward to extract a maximum edge biclique [88, Section 6.2].

It is proved that a bipartite graph has a perfect edge-without-vertex erasing order if and only if it is chordal bipartite [41, Theorem 13.17]. Therefore for chordal bipartite graphs one can compute the maximum edge biclique from a pewve order and the pewve order may be computed using their totally balanced biadjacency matrix as follows. Let $\mathbf{X} \in \{0,1\}^{m\times n}$ be a totally balanced matrix in a $\mathbf{\Gamma}$-free ordering and without loss of generality assume that it has no 0 rows and columns. Let σ be an ordering of $\text{supp}_1(\mathbf{X})$ in which the 1s are listed from top row to bottom and within each row from left to right. So for instance, matrix $\mathbf{J}_2$ would have the order $\sigma = \{(1,1),(1,2),(2,1),(2,2)\}$. Then σ is a pewve order, as for each $(i,j) \in \text{supp}_1(\mathbf{X})$ the 1s that are in column j below (i,j), (so $(\ell,j) \in \text{supp}_1(\mathbf{X})$ for any $\ell > i$) and the 1s that are in row i to the right of (i,j) (so $(i,k) \in \text{supp}_1(\mathbf{X})$ for any $k > j$) are in a $\mathbf{J}_2$ submatrix as $\mathbf{X}$ is $\mathbf{\Gamma}$-free. In the bipartite setting $\mathcal{B}(\mathbf{X})$, this shows that edge (i,j) is bisimplicial in the subgraph in which edges that come before (i,j) in σ are deleted.

2.3.4 Step number and jump number

The jump and step number of a partially ordered set (poset) are usually defined in the context of extending the poset into a total order. Here we give a simple equivalent graph theoretic definition. Given a directed acyclic graph $\vec{G}$, the jump number $jn(\vec{G})$ of $\vec{G}$ is the minimum number of arcs (directed edges) to be added to $\vec{G}$ so that it contains a directed Hamiltonian path (a directed path that visits each vertex exactly once) without creating any directed cycles in $\vec{G}$. Then the Hamiltonian path that is created consists of original arcs of $\vec{G}$ which are called *steps* and the arcs that are added in which are called *jumps*. The maximum number of steps in a Hamiltonian path so created is called the *step number* $sn(\vec{G})$ of $\vec{G}$. We have the following relation between the step and jump number of $\vec{G}$,

$$sn(\vec{G}) + jn(\vec{G}) = |V(\vec{G})| - 1.$$

Let $G = (V_1, V_2, E)$ be a bipartite graph. Let $\vec{E}$ be an orientation of the edge set of G so that each arc is directed from V_1 to V_2 and define $\vec{G} = (V_1, V_2, \vec{E})$ which is clearly a directed acyclic graph. Chaty and Chein [17] show that the cardinality of a maximum alternating cycle-free matching of a bipartite graph G is equal to the step number of $\vec{G}$. In chordal bipartite graphs, alternating cycle-free matchings are just alternating C_4-free matchings. Therefore, the isolation number of a totally balanced matrix $\mathbf{X}$ is equal to the step number of the above described simple orientation of $\mathcal{B}(\mathbf{X})$.

Chapter 3

Theory of Firmness

In this chapter, we present a detailed review of the literature on the theory of firmness, which we believe cannot be found in any textbook or paper. We carefully go through the definitions of firmness, superfirmness and rectangle cover graphs and illustrate them with examples. We also present some basic matrix operations that preserve the Boolean rank and isolation number. Then we explore the parallels between polynomial time algorithms for perfect graphs and firm matrices. Finally, we explore in great depth all so far known classes of firm and superfirm matrices.

3.1 Preliminaries

Let $\mathbf{X}$ be an $m \times n$ binary matrix. Following [77] we define the *rectangle cover graph* of $\mathbf{X}$, $\mathcal{G}(\mathbf{X})$, whose vertex set corresponds to the 1s of $\mathbf{X}$ and two vertices are adjacent in $\mathcal{G}(\mathbf{X})$ if the corresponding 1s in $\mathbf{X}$ belong to a common rectangle:

$$\begin{aligned}
&V(\mathcal{G}) = \mathrm{supp}_1(\mathbf{X}),\\
&E(\mathcal{G}) = \{[(i_1, j_1), (i_2, j_2)] : (i_1, j_1), (i_1, j_2), (i_2, j_1), (i_2, j_2) \in \mathrm{supp}_1(\mathbf{X})\}.
\end{aligned}$$

The *line graph* of a graph G with edge set E is defined to be the graph with vertex set E, where two vertices $e, f \in E$ are adjacent if e and f share a vertex in G. The rectangle cover graph $\mathcal{G}(\mathbf{X})$ may be thought of as a modified line graph of the bipartite representation $\mathcal{B}(\mathbf{X})$ of $\mathbf{X}$, in which vertices are also adjacent if the corresponding edges are contained in an induced C_4 subgraph. We adopt the convention that $\mathcal{G}(\mathbf{X})$ is drawn so that its vertices are in the same position where the corresponding 1s are in $\mathbf{X}$. As 1s of $\mathbf{X}$ and vertices of $\mathcal{G}(\mathbf{X})$ are in direct correspondence, we may simply refer to the vertices of $\mathcal{G}(\mathbf{X})$ as 1s of $\mathbf{X}$ or vice-versa.

Example 3.1.1. *Let* $\mathbf{D}_4$ *be the matrix below,*

$$\mathbf{D}_4 = \begin{bmatrix} 1 & 1 & & \\ 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 \\ & 1 & 1 & \end{bmatrix}. \tag{3.1.1}$$

(We have already seen this matrix in Equation (2.3.1) *in a different row and column order, however in the rest of the thesis, we will always use the ordering as given in Equation* (3.1.1) *above).*

The rectangle cover graph $\mathcal{G}(\mathbf{D}_4)$ *drawn according to our drawing convention is shown in Figure 3.1.*

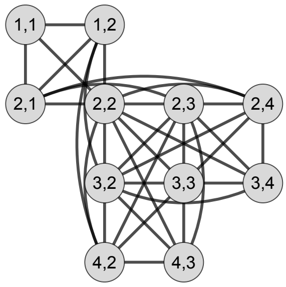


Figure 3.1: The rectangle cover graph $\mathcal{G}(\mathbf{D}_4)$

By our drawing convention, we can see that a rectangle cover graph can have three types of edges: (1) a *vertical edge* is between two vertices in the same column, (2) a *horizontal edge* is between two vertices in the same row, (3) a *diagonal edge* is between two vertices that are not in the same row nor in the same column.

Lubiw observed in [77], that working with $\mathcal{G}(\mathbf{X})$ gives us the advantage of translating the rectangle cover problem on $\mathbf{X}$ into a clique cover problem and the maximum isolated set problem into a maximum independent set problem. It is easy to see that two vertices of $\mathcal{G}(\mathbf{X})$ are not adjacent if and only if the corresponding 1s of $\mathbf{X}$ cannot be covered by a common rectangle.

Lemma 3.1.2. $S \subseteq \text{supp}_1(\mathbf{X})$ *is an independent set of* $\mathcal{G}(\mathbf{X})$ *if and only if it is an isolated set of* $\mathbf{X}$*.*

This equivalence between isolated sets of $\mathbf{X}$ and independent sets of $\mathcal{G}(\mathbf{X})$ shows that we may obtain a maximum isolated set of $\mathbf{X}$ by computing a maximum independent set of $\mathcal{G}(\mathbf{X})$. A similar result may be stated about maximal rectangles of $\mathbf{X}$.

Lemma 3.1.3. *[77, Claim 6.1.] K is a maximal clique of $\mathcal{G}(\mathbf{X})$ if and only if it is a maximal rectangle of $\mathbf{X}$.*

Proof. If $I \times J$ is a maximal rectangle of $\mathbf{X}$, then $I \times J = K$ is clearly a clique of $\mathcal{G}(\mathbf{X})$. Suppose K is not maximal, so there exist $(\ell, k) \in \text{supp}_1(\mathbf{X}) \setminus K$ such that $(i, j) \cup K$ is another clique. As (ℓ, k) is then adjacent to every $(i, j) \in K$, we have $x_{\ell k} = x_{\ell j} = x_{ik} = x_{ij} = 1$ for all $(i, j) \in K$. But then $(I \cup \{\ell\}) \times (J \cup \{k\})$ is a larger rectangle of $\mathbf{X}$, a contradiction. Hence, K is maximal.

Conversely, let K be a maximal clique of $\mathcal{G}(\mathbf{X})$. Let $I = \{i : (i, j) \in K\}$ and $J = \{j : (i, j) \in K\}$. We need to show that $I \times J \subset \text{supp}_1(\mathbf{X})$. Let $i_1 \in I$ and $j_2 \in J$ be arbitrary. Since $i_1 \in I$, there is some j_1 for which $(i_1, j_1) \in \text{supp}_1(\mathbf{X})$. If $j_1 = j_2$, then $(i_1, j_2) \in \text{supp}_1(\mathbf{X})$. Otherwise for j_2 there is some $i_2 \in I$ such that $(i_2, j_2) \in \text{supp}_1(\mathbf{X})$. Similarly, if $i_2 = i_1$ then $(i_1, j_2) \in \text{supp}_1(\mathbf{X})$. Otherwise, $(i_1, j_1), (i_2, j_2) \in K$ are adjacent vertices in $\mathcal{G}(\mathbf{X})$ with $i_1 \neq i_2$ and $j_1 \neq j_2$, which shows that $(i_1, j_2), (i_2, j_1) \in \text{supp}_1(\mathbf{X})$. Therefore, $K = I \times J$ is a rectangle. And $I \times J$ is a maximal rectangle, as any larger rectangle that would contain $I \times J$ would imply the existence of a clique containing K. □

The above lemma shows that there is an equivalence between maximal cliques of $\mathcal{G}(\mathbf{X})$ and maximal rectangles of $\mathbf{X}$, hence we may obtain a minimum rectangle cover of $\mathbf{X}$ by computing a minimum clique cover of $\mathcal{G}(\mathbf{X})$ that uses maximal cliques. It is important to note that Lemma 3.1.3 is only about *maximal* cliques of $\mathcal{G}(\mathbf{X})$. While it is easy to see that any rectangle of $\mathbf{X}$ is a clique of $\mathbf{X}$, not every clique of $\mathcal{G}(\mathbf{X})$ corresponds to a rectangle.

Example 3.1.4. *The rectangle cover graph of $\mathbf{J}_2$ is K_4. The subgraph of $\mathcal{G}(\mathbf{J}_2)$ induced by any three vertices is K_3 which does not correspond to any rectangle of $\mathbf{J}_2$.*

A non-maximal clique of $\mathcal{G}(\mathbf{X})$ may not correspond to a rectangle of $\mathbf{X}$, because not every subgraph of $\mathcal{G}(\mathbf{X})$ corresponds to a submatrix of $\mathbf{X}$. One can also see that deleting a vertex of $\mathcal{G}(\mathbf{X})$ retains all edges of $\mathcal{G}(\mathbf{X})$ that are not adjacent to the deleted vertex, while the deceivingly similar action of replacing a 1 by a 0 in $\mathbf{X}$ can remove edges from $\mathcal{G}(\mathbf{X})$ that are not necessarily adjacent to the vertex that corresponds to the deleted 1.

Example 3.1.4 (Continued)**.** *Deleting vertex $(1, 1)$ from $\mathcal{G}(\mathbf{J}_2)$ we get K_3, while replacing the 1 at $(1, 1)$ by a 0 in $\mathbf{J}_2$ we get $\mathbf{X}' = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right]$, and $\mathcal{G}(\mathbf{X}')$ is just a path on three nodes, as edge $[(2, 1), (1, 2)]$ disappears by setting $(1, 1)$ to 0.*

On the other hand, one can see that any submatrix of $\mathbf{X}$ formed by taking rows I and column J corresponds to a induced subgraph of $\mathcal{G}(\mathbf{X})$ obtained by deleting all vertices (i, j) for which $i \notin I$ or $j \notin J$. We summarise these observations below.

Observation 3.1.5. *If $\mathbf{X}'$ is a submatrix of $\mathbf{X}$, then $\mathcal{G}(\mathbf{X}')$ is a induced subgraph of $\mathcal{G}(\mathbf{X})$. But not every induced subgraph of $\mathcal{G}(\mathbf{X})$ corresponds to a submatrix of $\mathbf{X}$.*

Lastly, we mention that it would be an interesting research direction to understand whether for a general input graph G one could test efficiently if G is a rectangle cover graph. Sadly, we do not know if this is possible or not.

3.1.1 Graph perfection

In this section, we give a brief introduction into perfect graphs and some operations that preserve graph perfection. Recall that $\alpha(G), \omega(G), \chi(G)$ and $\theta(G)$ denote the independence, clique, chromatic and the clique cover number of a graph G. Furthermore, recall that $\alpha(G) \leq \theta(G)$, $\omega(G) \leq \chi(G)$ and $\alpha(\overline{G}) = \omega(G)$, $\theta(\overline{G}) = \chi(G)$. G is said to be *perfect* if $\alpha(G') = \theta(G')$ holds for all induced subgraphs G' of G, including G. This is not a standard definition, as one usually describes perfection using $\omega(G)$ and $\chi(G)$. However, by the weak perfect graph theorem that we state below, our definition is correct too. Let us illustrate this by stating some important theorems about perfect graphs. To *replicate a vertex v* of a graph is to introduce a new vertex v', connect it to v and then connect it to all the neighbours of v. In 1972, Lovász proved that vertex replication preserves perfection.

Lemma 3.1.6 (Replication Lemma [72]). *If G is a graph obtained by replicating a vertex of a perfect graph, then G is perfect. In particular, vertex replication preserves perfection.*

Using the Replication Lemma, Lovász proved that graph complementation also preserves perfection. This theorem is one of the most important results in graph theory and is often referred to as the Weak Perfect Graph Theorem.

Theorem 3.1.7 (Weak Pefect Graph Theorem [72]). *A graph is perfect if and only if its complement is perfect.*

Furthermore, Lovász later proved a characterisation of perfect graphs which implies the weak perfect graph theorem and will be used in Section 3.2 to illustrate a polynomial time algorithm to compute a minimum clique cover of perfect graphs.

Theorem 3.1.8 (Lovász' Characterisation of Perfection [71])**.** *A graph G is perfect if and only if $\omega(H) \cdot \alpha(H) \geq |V(H)|$ for every induced subgraph H of G.*

It had been an open question for a long time to characterise perfect graphs in terms of forbidden subgraphs. A graph G is said to be *minimally imperfect* if $\alpha(G') = \theta(G')$ holds for all induced proper subgraphs G' of G, but not for G. We say that a graph G *has or contains a graph H* if G has an induced subgraph which is isomorphic to H. If G does not have a graph H as an induced subgraph then we say that G is *H-free.* It is clear that perfect graphs are exactly the ones that do not contain any minimally imperfect graphs. Therefore, the real difficulty is to fully characterise minimally imperfect graphs.

A *chord* of a cycle is an edge that connects two non-consecutive vertices of the cycle. A *hole* is a chordless cycle of at least four vertices. An *antihole* is the complement of a hole. An *odd (anti)hole* is an (anti)hole with an odd number of vertices. One can verify that odd holes and odd antiholes are minimally imperfect. It was not until 2006, when Chudnovsky et al. proved a deep result in which they show that odd holes and odd antiholes are the only minimally imperfect graphs. This result is often referred to as the Strong Perfect Graph Theorem.

Theorem 3.1.9 (Strong Perfect Graph Theorem [19])**.** *A graph is perfect if and only if it has no odd hole and no odd antihole.*

We will make extensive use of this powerful theorem in the rest of the thesis, often in the context when we want to show that a rectangle cover graph is perfect if it does not have any odd holes and odd antiholes.

Let us present one more operation that preserves perfection which will be used in Section 3.3.2 to illustrate that Lubiw's L-sum preserves superfirmness. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \emptyset$, which both contain a clique of size $k > 0$, say $\mathcal{K}_1$ and $\mathcal{K}_2$. The *clique-sum* of G_1 and G_2 is a graph $G = (V, E)$ formed by identifying vertices of $\mathcal{K}_1$ with vertices of $\mathcal{K}_2$ via a bijection $f : \mathcal{K}_1 \to \mathcal{K}_2$ and then gluing together G_1 and G_2 at their cliques $\mathcal{K}_1$ and $\mathcal{K}_2$ to form a single shared clique. So that the vertices of G are given by

$$V = V_1 \cup (V_2 \setminus \mathcal{K}_2),$$

and the edge set of G is given by

$$E = E_1 \ \cup \ (E_2 \setminus E(\mathcal{K}_2)) \ \cup \ \{(i, j) : \ i \in \mathcal{K}_1, \ j \in V_2 \setminus \mathcal{K}_2 \text{ and } (f(i), j) \in E_2\}).$$

Berge shows that clique sum preserves perfection.

Lemma 3.1.10 (Clique-sum Lemma [9])**.** *If G is a graph obtained by the clique sum of two perfect graphs, then G is perfect. In particular, the clique-sum operation preserves perfection.*

3.1.2 Superfirmness

Recall that $\mathbf{X}$ is said to be firm if $\mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}')$ for all submatrices $\mathbf{X}'$ of $\mathbf{X}$, including $\mathbf{X}$. Following [77], let $\mathbf{X}$ be called *superfirm* if $\mathcal{G}(\mathbf{X})$ is perfect. How do firmness and superfirmness relate to each other? By Observation 3.1.5, every submatrix of $\mathbf{X}$ corresponds to an induced subgraph of $\mathcal{G}(\mathbf{X})$, so if $\mathbf{X}$ is superfirm then $\mathbf{X}$ is firm. On the other hand, since not every induced subgraph of $\mathcal{G}(\mathbf{X})$ corresponds to a submatrix of $\mathbf{X}$, if $\mathbf{X}$ is firm, it does not necessarily need to be superfirm as the inequality $\alpha(H) = \theta(H)$ needs only hold for induced subgraphs H of $\mathcal{G}(\mathbf{X})$ which correspond to a submatrix of $\mathbf{X}$. Indeed, the following example of Lubiw shows that superfirmness is a strictly stronger requirement than firmness and superfirm matrices are a strict subset of firm matrices.

Example 3.1.1 (Continued)**.** *The rectangle cover graph $\mathcal{G}(\mathbf{D}_4)$ contains a 5-hole as indicated in Figure 3.2, hence $\mathcal{G}(\mathbf{D}_4)$ is not perfect, and $\mathbf{D}_4$ is not superfirm. On the other hand, $\mathbf{D}_4$ is a row-column interval matrix, hence it is firm by Chaiken et al's theorem [14].*

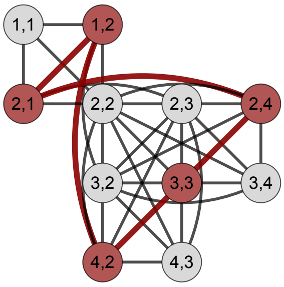


Figure 3.2: A 5-hole in the rectangle cover graph of firm matrix $\mathbf{D}_4$

In Example 3.1.4 we have seen that replacing a 1 of $\mathbf{X}$ by a 0 can remove edges from $\mathcal{G}(\mathbf{X})$ that are not necessarily adjacent to the vertex that corresponds to the deleted 1. To overcome this and to be able to represent any subgraph of $\mathcal{G}(\mathbf{X})$ in some sort of matrix form, let us turn our attention to generalised binary matrices.

Recall from Section 1.3, that a matrix over $\{0, 1, ?\}$ is called a generalised binary matrix. For a generalised binary matrix $\mathbf{Y}$, a submatrix $I \times J$ is a rectangle if $I \times J \subseteq \mathrm{supp}_1(\mathbf{Y}) \cup \mathrm{supp}_?(\mathbf{Y})$, while $S \subseteq \mathrm{supp}_1(\mathbf{Y})$ is an isolated set if no two elements of it can be covered by a common rectangle. Hence, ?'s can be used to form rectangles, cannot belong to an isolated set and are allowed but need not be covered in a rectangle covering.

Further recall, that for a standard binary matrix $\mathbf{X}$, for any $P \subset \operatorname{supp}_1(\mathbf{X})$, $\mathbf{X}^P$ is the generalised binary matrix obtained by setting entries $(i, j) \in P$ to ?'s. Let us define the rectangle cover graph of the generalised binary matrix $\mathbf{X}^P$ as the induced subgraph of $\mathcal{G}(\mathbf{X})$ obtained by deleting vertices in P.

Example 3.1.1 (Continued)**.** *Let* $P = \{(1,1),(2,2),(2,3),(3,2),(3,4),(4,3)\} \subset \operatorname{supp}_1(\mathbf{D}_4)$. *The generalised binary matrix* $\mathbf{D}_4^P$ *and its rectangle cover graph* $\mathcal{G}(\mathbf{D}_4^P)$ *are shown in Figure 3.3. Observe that the induced subgraph* $\mathcal{G}(\mathbf{D}_4^P)$ *of* $\mathcal{G}(\mathbf{D}_4)$ *which is just a* 5*-hole, does not correspond to any submatrix of* $\mathbf{D}_4$*, hence using generalised binary matrices is the only way to 'assign' a matrix to it.*

$$\mathbf{D}_4^P = \begin{bmatrix} ? & 1 & & \\ 1 & ? & ? & 1 \\ & ? & 1 & ? \\ & 1 & ? & \end{bmatrix} \qquad \mathcal{G}(\mathbf{D}_4^P) =$$

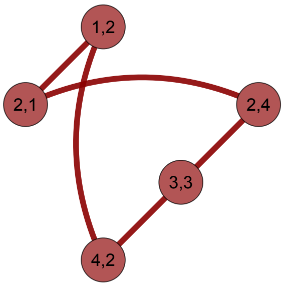


Figure 3.3: Matrix $\mathbf{D}_4$ and its rectangle cover graph $\mathcal{G}(\mathbf{D}_4)$

Therefore, using generalised binary matrices we are able to represent every induced subgraph of rectangle cover graphs in matrix form. Furthermore, as Lubiw already observed in [77], superfirmness of a standard binary matrix $\mathbf{X}$ is equivalent to the property that for every $P \subset \operatorname{supp}_1(\mathbf{X})$ we have $\mathfrak{i}(\mathbf{X}^P) = \mathfrak{br}(\mathbf{X}^P)$.

Firmness can also be defined for generalised binary matrices. A generalised binary matrix $\mathbf{Y}$ is said to be *firm* if $\mathfrak{i}(\mathbf{Y}') = \mathfrak{br}(\mathbf{Y}')$ holds for any submatrix $\mathbf{Y}'$ of $\mathbf{Y}$, including $\mathbf{Y}$. Firmness of a generalised binary matrix should not be confused with the superfirmness of the standard binary matrix $\mathbf{X}$ for which we can write $\mathbf{Y} = \mathbf{X}^P$ for some $P \subset \operatorname{supp}_1(\mathbf{X})$ as superfirmness of $\mathbf{X}$ is a stronger requirement than the firmness of $\mathbf{Y}$.

3.1.3 Basic matrix operations

Recall that the Boolean column space $BCS(\mathbf{X})$ of a binary matrix $\mathbf{X}$ is the set containing the zero vector and all vectors that can be obtained as the Boolean sum of columns of $\mathbf{X}$. For any $\boldsymbol{v} \in BCS(\mathbf{X})$ we say that the matrix $[\mathbf{X}, \boldsymbol{v}]$ in which vector $\boldsymbol{v}$ is appended to $\mathbf{X}$ is created by *a Boolean column space extension.* Phelps [94] proves that Boolean column (or row) space extension preserves the Boolean rank and isolation number.

Lemma 3.1.11. *[94, Theorem 3.5] If* $\mathbf{X}'$ *is obtained by* a Boolean column space extension *of* $\mathbf{X}$ *then* $\mathfrak{br}(\mathbf{X}') = \mathfrak{br}(\mathbf{X})$ *and* $\mathfrak{i}(\mathbf{X}') = \mathfrak{i}(\mathbf{X})$.

We were interested if this result could be strengthened to show that Boolean space extensions preserve firmness. Unfortunately the answer is no as the below counterexample shows that we may obtain the non-firm submatrix $\bar{\mathbf{I}}_4$ starting from a firm matrix.

Example 3.1.12. *Let* $\mathbf{X}$ *be the matrix shown below and let the vector* $\boldsymbol{v}$ *be the Boolean sum of the first three columns of* $\mathbf{X}$. *Append* $\boldsymbol{v}$ *to* $\mathbf{X}$ *to obtain matrix* $\mathbf{X}'$ *as shown below. One can verify that* $\mathbf{X}$ *is a firm matrix while the submatrix indexed by* $\{1,2,3,4\} \times \{4,5,6,7\}$ *of* $\mathbf{X}'$ *is matrix* $\bar{\mathbf{I}}_4$ *which is a non-firm matrix.*

$$\mathbf{X} = \begin{bmatrix} 1 & & & & 1 & 1 \\ & 1 & & 1 & & 1 \\ & & 1 & 1 & 1 & \\ & & & 1 & 1 & 1 \end{bmatrix}, \ \boldsymbol{v} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \bigvee_{j=1}^{3} \mathbf{X}_{:,j}, \ \mathbf{X}' = \begin{bmatrix} 1 & & & & 1 & 1 & 1 \\ & 1 & & 1 & & 1 & 1 \\ & & 1 & 1 & 1 & & 1 \\ & & & 1 & 1 & 1 & \end{bmatrix}$$

Nonetheless, a version of the Boolean space extension restricted to the all 1s row and column can be shown to preserve firmness. We say the matrix $[\mathbf{1}, \mathbf{X}]$ is obtained by *extending* $\mathbf{X}$ *with an all* 1*s column.*

Lemma 3.1.13. *Extending a firm matrix* $\mathbf{X}$ *with an all* 1*s row or column preserves firmness.*

Proof. Let $\mathbf{X}$ be a firm matrix and let $\mathbf{Y} = [\mathbf{1}, \mathbf{X}]$. Let $\mathbf{Y}' = [\mathbf{1}, \mathbf{X}']$ be any submatrix of $\mathbf{Y}$ which includes some entries from the all 1s column, and $\mathbf{X}'$ be the corresponding submatrix of $\mathbf{X}$. Then as $\mathbf{X}$ is firm, $\mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}')$. In addition, since $\mathbf{X}'$ is submatrix of $\mathbf{Y}'$, we have $\mathfrak{br}(\mathbf{X}') \leq \mathfrak{br}(\mathbf{Y}')$ and $\mathfrak{i}(\mathbf{X}') \leq \mathfrak{i}(\mathbf{Y}')$.

If $\mathbf{X}'$ has no row of all 0s then the rectangles in a minimum rectangle cover of $\mathbf{X}'$ can be extended to cover all the 1s in column $\mathbf{1}$, so $\mathfrak{br}(\mathbf{Y}') = \mathfrak{br}(\mathbf{X}')$ and $\mathfrak{i}(\mathbf{Y}') = \mathfrak{br}(\mathbf{Y}')$.

If $\mathbf{X}'$ has a row of all 0s, then the 1 in that row from column $\mathbf{1}$ is isolated from all other 1s of $\mathbf{X}'$, hence $\mathfrak{i}(\mathbf{X}') + 1 \leq \mathfrak{i}(\mathbf{Y}')$. Adding one more rectangle to a minimum cover of $\mathbf{X}'$ to cover that 1, we get a feasible cover of $\mathbf{Y}'$, so $\mathfrak{br}(\mathbf{Y}') \leq \mathfrak{br}(\mathbf{X}') + 1$ and $\mathfrak{i}(\mathbf{Y}') = \mathfrak{br}(\mathbf{Y}')$. □

Could it be that the all 1s row extension also preserves superfirmness? The below example shows that it does not.

Example 3.1.14. *Let* $\mathbf{H}_3 := [\mathbf{1}, \mathbf{C}_3]$ *be obtained by extending the* 3×3 *cycle matrix with* $\mathbf{1}$. $\mathbf{C}_3$ *is a superfirm matrix as* $\mathcal{G}(\mathbf{C}_3)$ *is a* 6*-hole. By Lemma 3.1.13,* $\mathbf{H}_3$ *is firm. On the other hand, Figure 3.4 shows that* $\mathbf{H}_3$ *is not superfirm as* $\mathcal{G}(\mathbf{H}_3)$ *contains three* 5*-holes.*

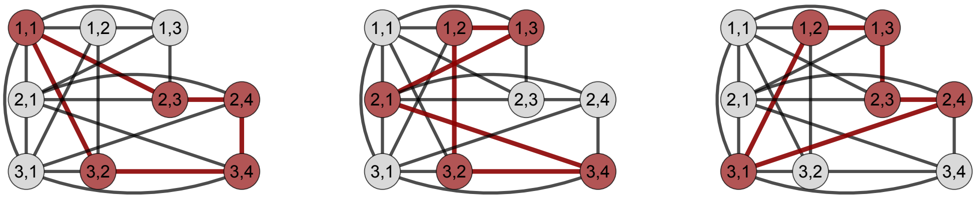


Figure 3.4: The three 5-holes in $\mathcal{G}(\mathbf{H}_3)$

Therefore, extending with an all 1s row or column preserves firmness but does not preserve superfirmness.

Let us now turn our attention to some classical matrix operations. Perhaps, the simplest operation that may be applied to two binary matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ is the direct sum operation, which forms a block diagonal matrix $\left[\begin{smallmatrix}\mathbf{X}_1 & \\ & \mathbf{X}_2\end{smallmatrix}\right]$. As the diagonal blocks share no common rectangles, one can easily see the direct sum operation preserves superfirmness and firmness.

The second natural operation to question is the Boolean matrix product. However, the below example shows that the Boolean matrix product cannot preserve firmness nor superfirmness because the non-firm matrix $\bar{\mathbf{I}}_4$ can be factorised as,

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & & \\ 1 & & 1 & \\ & 1 & & 1 \\ & & 1 & 1 \end{bmatrix} \circ \begin{bmatrix} & & 1 & 1 \\ & 1 & & 1 \\ 1 & & 1 & \\ 1 & 1 & & \end{bmatrix},$$

where the factor matrices are just permutations of the superfirm matrix $\mathbf{C}_4$.

The last simple matrix operation that we consider is the Kronecker product. Caen et al. [29],[86, pg. 48] prove that the following inequalities govern the Boolean rank and isolation number of the Kronecker product $\mathbf{X} \otimes \mathbf{Y}$,

$$\begin{aligned} \max\left\{\mathfrak{i}(\mathbf{X})\mathfrak{br}(\mathbf{Y}), \mathfrak{br}(\mathbf{X})\mathfrak{i}(\mathbf{Y})\right\} \leq \quad & \mathfrak{br}(\mathbf{X} \otimes \mathbf{Y}) & \leq \mathfrak{br}(\mathbf{X})\mathfrak{br}(\mathbf{Y}), \\ \mathfrak{i}(\mathbf{X})\mathfrak{i}(\mathbf{Y}) \leq \quad & \mathfrak{i}(\mathbf{X} \otimes \mathbf{Y}) & \leq \min\left\{\mathfrak{i}(\mathbf{X})\mathfrak{br}(\mathbf{Y}), \mathfrak{br}(\mathbf{X})\mathfrak{i}(\mathbf{Y})\right\}. \end{aligned}$$

This shows that for any weakly firm matrices $\mathbf{X}$ and $\mathbf{Y}$, their Kronecker product is also weakly firm. However, it turns out that Kronecker product does not preserve firmness a the Kronecker product of two firm matrices below results in a matrix that has $\overline{\mathbf{I}}_4$ as a submatrix,

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Similarly, the Kronecker product of superfirm matrices $\mathbf{\Gamma}$ and $\mathbf{D}_3$ results in a matrix whose rectangle cover graph contains several 5-holes, the non-superfirm $\mathbf{H}_3$ submatrix which contains three 5-holes as shown in Figure 3.4 is highlighted,

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right].$$

Therefore, the Kronecker product does not preserve superfirmness either.

3.2 Algorithms for perfect graphs

In this section, we detail Grötschel et al.'s [46] polynomial time algorithm to compute a maximum independent set and minimum clique cover of perfect graphs. The Lovász $\vartheta(G)$ number [73] is a real number that is sandwiched between the clique cover and the independence number for every graph, so for any graph $G = (V, E)$

$$\alpha(G) \leq \vartheta(G) \leq \theta(G).$$

For any $\epsilon > 0$, $\vartheta(G)$ can be computed to within ϵ precision by solving a semidefinite program in time polynomial in $|V(G)|$ and $\log \frac{1}{\epsilon}$ [45]. Let us denote the value of $\vartheta(G)$ computed to ϵ precision by $\vartheta_\epsilon(G)$, so we have $\vartheta_\epsilon(G) \in [\vartheta(G) - \epsilon, \vartheta(G) + \epsilon]$. If G is weakly perfect, then we can compute $\alpha(G) = \theta(G)$ in polynomial time. This is because, for weakly perfect graphs $\vartheta(G)$ is an integer number and it suffices to solve the semidefinite program with precision $\epsilon < \frac{1}{2}$ and then can get $\alpha(G) = \theta(G)$ by rounding $\vartheta_\epsilon(G)$ to the nearest integer. Let us denote $\vartheta_\epsilon(G)$ rounded to the nearest integer by $\lfloor \vartheta_\epsilon(G) \rceil$.

In addition, Grötschel et al. [46] show that if G is perfect, not only the numbers $\alpha(G)$, $\theta(G)$ can be computed, but a maximum independent set and minimum clique partition too. We illustrate this following [67, Chapter 4]. Algorithm 1 shows how a maximum independent set can be computed for a perfect graph. The idea is to remove a vertex at each iteration to get a subgraph G_i of G, compute $\alpha(G_i)$ by computing $\vartheta_\epsilon(G_i)$ with precision $\epsilon < \frac{1}{2}$, rounding it to the nearest integer $\lfloor \vartheta_\epsilon(G_i) \rceil$ and check whether the removal of the vertex decreases $\alpha(G)$. If the vertex's removal decreases α, then that vertex must be included in a maximum independent set and so it is left in the graph, otherwise the vertex is permanently removed. After at most $|V(G)|$ iterations, the subgraph G_n that remains is an independent set of size $\alpha(G)$.

Algorithm 1: MaxIndependentSet(G, ϵ)

```
Input: $G = (V, E)$, $\epsilon \in (0, \frac{1}{2})$.
Set $G_0 := G$, compute $\vartheta_0 := \lfloor \vartheta_\epsilon(G_0) \rceil$ and order nodes $v_1, \dots, v_n$.
for $i \in [n]$ do
    Compute $\vartheta_i := \lfloor \vartheta_\epsilon(G_{i-1} \setminus v_i) \rceil$.
    if $\vartheta_i = \vartheta_0$ then
        Set $G_i = G_{i-1} \setminus v_i$.
    else
        Set $G_i = G_{i-1}$.
Output: $G_n$.
```

To compute a minimum clique partition of a perfect graph G, somewhat more work needs to be done. The method presented in Algorithm 2 is based on the observation that for a perfect graph G it suffices to find a clique K of G that intersects all maximum independent sets of G and then we can recursively partition $G \setminus K$ into cliques. Such a clique exists because any clique K that is a member of a minimum clique partition of G satisfies $\theta(G \setminus K) = \theta(G) - 1 = \alpha(G) - 1$.

A clique intersecting all maximum independent sets of G can be found iteratively. Keep a list $\mathcal{L} = \{S_1, \dots\}$ of maximum independent sets of G that have already been computed and at each step compute a clique K that intersects all sets in $\mathcal{L}$. If $\alpha(G \setminus K) = \lfloor \vartheta_\epsilon(G \setminus K) \rceil < \lfloor \vartheta_\epsilon(G) \rceil = \alpha(G)$, then K intersects all maximum independent sets of G, otherwise a maximum independent set of $G \setminus K$ of size $\alpha(G)$ is added to the list $\mathcal{L}$ and the process is iterated.

Algorithm 2: MinCliquePartition(G, ϵ)

```
Input: $G = (V, E)$, $\epsilon \in (0, \frac{1}{2})$.
Compute $S_1 :=$ MaxIndependentSet$(G, \epsilon)$ and set $\mathcal{L} := \{S_1\}$, $\mathcal{C} := \{\}$.
Call CP$(G, \mathcal{L}, \mathcal{C})$.
Output: $\mathcal{C}$.

Algorithm CP$(G, \mathcal{L}, \mathcal{C})$:
Compute a clique $K$ that intersects all independent sets in $\mathcal{L}$.
if $\lfloor \vartheta_\epsilon(G \setminus K) \rceil = \lfloor \vartheta_\epsilon(G) \rceil$ then
    Update $\mathcal{L} := \mathcal{L} \cup \{$MaxIndependentSet$(G \setminus K)\}$
    Call CP$(G, \mathcal{L}, \mathcal{C})$.
else
    Update $\mathcal{C} := \mathcal{C} \cup \{K\}$
if $V(G \setminus K) \neq \emptyset$ then
    Call CP$(G \setminus K, \{S \setminus K : S \in \mathcal{L}\}, \mathcal{C})$.
```

In order for this algorithm to work, one needs to be able to compute a clique K that intersects all independent sets in list $\mathcal{L} = \{S_1, \ldots, S_t\}$. For this let H be a subgraph of G induced by $\cup_{i=1}^{t} S_i$ in which each vertex is replicated as many times as it appears among S_i's. Then by the Vertex Replication Lemma 3.1.6 H is perfect, so $\omega(H) = \chi(H)$. We have $\chi(H) \leq t$, because each vertex that has been replicated k times for some $k \leq t$, belongs to k independent sets S_i and can be coloured accordingly. In addition, we have $|V(H)| = t\alpha(H)$ and $\alpha(H) = \alpha(G)$. By Lovász' Characterisation of Perfection 3.1.8, H satisfies $|V(H)| = t\alpha(G) \leq \omega(H)\alpha(G)$. So $t \leq \omega(H)$ and a maximum clique of H intersects all S_i. Translating this clique back to G, by collapsing any vertex replication that has been made, we get a clique which may not have size t, but still intersects all S_i. Therefore, by computing MaxIndependentSet$(\overline{H}, \epsilon)$ to get a maximum clique of H and undoing vertex replication, we obtain the clique intersecting all independent sets in $\mathcal{L}$.

To see that this algorithm runs in polynomial time, observe that at each iteration we either add a new maximum independent set to $\mathcal{L}$ or remove a clique from G which reduces $\lfloor \vartheta_\epsilon(G) \rceil$ by exactly one. Next we argue that $|\mathcal{L}| \leq |V(G)|$. Let $\mathcal{L}$ contain t independent sets $S_1, \ldots, S_t$ at some iteration and let us define an affine space corresponding to these t independent sets as

$$L_t = \{\boldsymbol{x} \in \mathbb{R}^{|V|} : \sum_{i \in S} x_i = 1 \text{ for all } S \in \{S_1, \ldots, S_t\}\}.$$

Then the indicator vector $\boldsymbol{x}^K \in \{0,1\}^{|V|}$ ($x_i = 1 \iff i \in K$) of a clique K that intersects all $S_1, \ldots, S_t$ must satisfy $\boldsymbol{x}^K \in L_t$. If this clique K does not intersect a maximum independent set S_{t+1}, then $\sum_{i \in S_{t+1}} x_i^K = 0$ and $\boldsymbol{x}^K \in L_t \setminus L_{t+1}$, where $L_{t+1} = L_t \cap \{\boldsymbol{x} \in \mathbb{R}^{|V|} : \sum_{i \in S_{t+1}} x_i = 1\}$. Hence the dimension of the affine space decreases by at least one at each iteration when a new maximum independent set is added to $\mathcal{L}$ and thus $|\mathcal{L}| \leq |V(G)|$ and the total number of iterations is bounded by $|V(G)|$.

3.2.1 Algorithms for firm and superfirm matrices?

Let us turn our attention back to matrices and let $\mathbf{X} \in \{0,1\}^{m \times n}$. By the properties of rectangle cover graphs, we know that $\mathfrak{br}(\mathbf{X}) = \theta(\mathcal{G}(\mathbf{X}))$ and $\mathfrak{i}(\mathbf{X}) = \alpha(\mathcal{G}(\mathbf{X}))$. $\mathcal{G}(\mathbf{X})$ may be built in $\mathcal{O}(m^2n^2)$ time, so if we can compute a maximum independent set of $\mathcal{G}(\mathbf{X})$ in polynomial time, we obtain a maximum isolated set of $\mathbf{X}$ in polynomial time. Similarly, if we can compute a minimum clique partition/cover of $\mathcal{G}(\mathbf{X})$, then each clique in the cover can be extended into a maximal clique in $\mathcal{O}(mn)$ time and hence we obtain a minimum rectangle cover of $\mathbf{X}$. When extending a clique to a maximal clique, different maximal cliques may be obtained based on the order the vertices are considered. However, any extension gives an optimal rectangle cover, perhaps just with a different overlap between the rectangles. Therefore, as one expects the main difficulty of computing a minimum rectangle cover and maximum isolated set is the computation of minimum clique covers and maximum independent sets.

If $\mathbf{X}$ is superfirm, $\mathcal{G}(\mathbf{X})$ is perfect and a minimum rectangle cover and a maximum isolated set can be computed in polynomial time using Algorithm 1 and 2. The question remains whether these algorithms could somehow be altered to work for firm matrices as well. By computing $\vartheta(\mathcal{G}(\mathbf{X}))$ with accuracy $\epsilon < \frac{1}{2}$, we can obtain numbers $\mathfrak{br}(\mathbf{X})$ and $\mathfrak{i}(\mathbf{X})$ for any weakly firm matrix (recall that weakly firmness is defined by $\mathfrak{br}(\mathbf{X}) = \mathfrak{i}(\mathbf{X})$). For firm matrices that are not superfirm, we have $\theta(H) = \alpha(H)$ for only those subgraphs H that correspond to a submatrix. Therefore, $\mathcal{G}(\mathbf{X})$ is a weakly perfect graph with some additional properties that do not immediately show whether the algorithms for perfect graphs can be successfully adapted. As the following example shows Algorithm 1 and 2 may fail for weakly perfect graphs.

Example 3.2.1. *Let $G = ([8], \{(1,2), E(C_7)\})$ be a graph which contains a 7-hole and exactly one other vertex (vertex 1) which is adjacent to only one vertex of the 7-hole (vertex 2). Then $\alpha(G) = \theta(G) = 4$. We run MaxIndependentSet(G, ϵ) on G, and examine vertices in label number order. After removing vertex 1 we compute $\vartheta(C_7) = \frac{7\cos(\pi/n)}{1+\cos(\pi/n)} = 3.317\ldots$ with some precision $\epsilon < \frac{1}{2}$. If $\epsilon = 0.4$ then $\vartheta_{0.4}(C_7) \in [2.917, 3.717]$, so we may get $\lfloor \vartheta_{0.4}(C_7) \rceil$ equal to 3 or 4. If $\lfloor \vartheta_{0.4}(C_7) \rceil = 4$ then the* *algorithm fails to detect that we removed a vertex which needs to be part of a maximum independent set.*

Let G be the graph in Figure 3.5 which is weakly perfect. G has 3 maximum independent sets of size 3 *given by* $\{1,6,8\},\{2,5,8\},\{3,5,7\}$ *and also a minimum clique cover of size* 3 *formed by the maximal cliques* $\{1,4,5\},\{2,6,7\},\{3,4,8\}$*. We call MinCliquePartition*(G,ϵ) *on G. Suppose we are lucky and successfully compute all maximum independent sets in* $\mathcal{L}$*. Then we find the maximal clique* $K=\{1,2,3\}$ *intersects all of them. After removing K, we get* $G\backslash K = C_5$*, and the algorithm fails as there is no clique of* C_5 *that intersects all of its maximum independent sets. Therefore, for weakly perfect graphs, even if a clique intersects all maximum independent sets, it cannot be assumed to be contained in a minimum clique cover.*

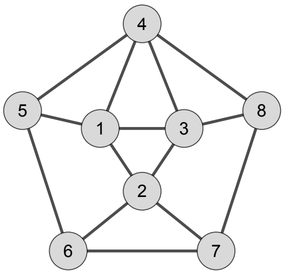


Figure 3.5: A weakly perfect graph for which Algorithm 2 can fail

So for weakly perfect graphs and weakly firm matrices Algorithm 1 and 2 do not work. However, using the submatrix hereditary property of firm matrices we suspect that one may be able to give a polynomial time algorithm to compute the maximum isolated set and minimum rectangle cover of a firm matrix as we can assume that we have a polynomial time oracle to compute $\mathfrak{br}(\mathbf{X}') = \mathfrak{i}(\mathbf{X}')$ for every submatrix $\mathbf{X}'$ of a firm matrix $\mathbf{X}$.

We also know that a full Boolean rank square submatrix of a firm matrix $\mathbf{X} \in \{0,1\}^{m\times n}$ may be extracted as follows (a full Boolean rank submatrix of $\mathbf{X}$ means a submatrix $\mathbf{Y}$ of $\mathbf{X}$ of dimension $\mathfrak{br}(\mathbf{X}) \times \mathfrak{br}(\mathbf{X})$ and of Boolean rank $\mathfrak{br}(\mathbf{Y}) = \mathfrak{br}(\mathbf{X})$). For each row $i \in [m]$ of $\mathbf{X}$, we can check whether dropping row i decreases $\mathfrak{i}(\mathbf{X})$ by computing the isolation number of the corresponding firm submatrix. Let $\mathbf{X} \setminus \mathbf{X}_{i,:}$ denote the matrix obtained from $\mathbf{X}$ by deleting row i. If $\mathfrak{i}(\mathbf{X} \setminus \mathbf{X}_{i,:}) = \mathfrak{i}(\mathbf{X})$, then row i can be permanently dropped. Otherwise, if $\mathfrak{i}(\mathbf{X} \setminus \mathbf{X}_{i,:}) < \mathfrak{i}(\mathbf{X})$ then row i contains a 1 that must be included in a maximum isolated set of $\mathbf{X}$ and the row is not deleted. Continuing this way, we obtain a matrix which has exactly $\mathfrak{i}(\mathbf{X})$-many rows. Then repeating this procedure on the columns of the output matrix, we can check for every column as well, whether deleting them decreases the isolation number. After this, we obtain a submatrix $\mathbf{Y}$ of $\mathbf{X}$ which is of dimension $\mathfrak{i}(\mathbf{X}) \times \mathfrak{i}(\mathbf{X})$ and has $\mathfrak{i}(\mathbf{Y}) = \mathfrak{i}(\mathbf{X})$. Therefore, each row and column of $\mathbf{Y}$ contains exactly one element from a maximum isolated set of $\mathbf{X}$.

At this point however, we are not sure how to extract a maximum isolated set from $\mathbf{Y}$. One may try to loop through all $(i, j) \in \mathrm{supp}_1(\mathbf{Y})$, and compute $\mathfrak{i}(\mathbf{Y} \setminus \mathbf{Y}_{i,:} \setminus \mathbf{Y}_{:,j})$, however we do not see a way to enforce that the 1s selected are not in a common rectangle.

Furthermore, since $\mathbf{Y}$ is of dimension $\mathfrak{br}(\mathbf{X}) \times \mathfrak{br}(\mathbf{X})$ with $\mathfrak{br}(\mathbf{Y}) = \mathfrak{br}(\mathbf{X})$, an optimal factorisation for it can be obtained by just taking its rows or columns. Then we may add back all the columns that were deleted from $\mathbf{X}$ to obtain a matrix $\mathbf{Z}$ of dimension $\mathfrak{i}(\mathbf{X}) \times n$ and an optimal factorisation for it is given by its rows. But from this point onwards, we do not know how to extend the factorisation when we keep adding the deleted rows of $\mathbf{X}$ back. We know, that none of those deleted rows can increase the Boolean rank, but we could not come up with a way to extend the factorisation for new rows. Therefore, we state the following conjecture.

Conjecture 3.2.2. *For every firm matrix, a minimum rectangle cover and a maximum isolated set can be computed in polynomial time.*

3.3 Known firm matrices

In this section, we give a list of matrices that are proved to be firm so far. For all the firm matrix classes that are known so far there is also a polynomial time algorithm to compute a minimum rectangle cover and a maximum isolated set, which also fuels our belief that it should be possible to compute a minimum rectangle cover and maximum isolated set in polynomial time for every firm matrix.

3.3.1 Linear matrices

The simplest class of firm matrices is one in which no 2×2 rectangles are allowed. A binary matrix which has no $\mathbf{J}_2$ submatrix is said to be *linear* [22].

Theorem 3.3.1. *[77, Lemma 5.2] Linear binary matrices are superfirm.*

Proof. Let $\mathbf{X}$ be a linear binary matrix. The bipartite representation $\mathcal{B}(\mathbf{X})$ of $\mathbf{X}$, then has no subgraph isomorphic to $K_{2,2}$, hence the rectangle cover graph $\mathcal{G}(\mathbf{X})$ of $\mathbf{X}$ is just the line graph of $\mathcal{B}(\mathbf{X})$. Since line graphs of bipartite graphs are perfect [41], $\mathcal{G}(\mathbf{X})$ is a perfect graph and therefore $\mathbf{X}$ is superfirm. □

By superfirmness we may use the general polynomial time algorithm for perfect graphs that is mentioned in the previous section to compute a minimum rectangle cover and maximum isolated set of a linear matrix $\mathbf{X}$. However, a simpler way is to observe that since the bipartite representation $\mathcal{B}(\mathbf{X})$ has no $K_{2,2}$ subgraphs, all of $\mathcal{B}(\mathbf{X})$'s bicliques are stars and any matching of $\mathcal{B}(\mathbf{X})$ is $K_{2,2}$-free. Hence, the isolation number of $\mathbf{X}$ is equal to the cardinality of a maximum matching in the bipartite representation $\mathcal{B}(\mathbf{X})$. Similarly, a minimum biclique cover of $\mathcal{B}(\mathbf{X})$ can be constructed by computing a minimum vertex cover to get a set of vertices $W \subset V(\mathcal{B}(\mathbf{X}))$ and then take the bicliques formed of the stars having a vertex in W along with all their neighbours. Since a minimum vertex cover and maximum matching can be computed in bipartite graphs in polynomial time [99, Chapter 16], for linear matrices the minimum rectangle cover and maximum isolated set are computable in polynomial time.

3.3.2 L-decomposable matrices

In this section we present two classes of superfirm matrices. Let us start first by describing an operation which is introduced and analysed by Lubiw in [77] in the context of binary matrices using their bipartite representation. In turn, Lubiw mentions that this operation comes from the more general method of split decomposition on graphs which is due to Cunningham and Edmonds [25].

L-sum. Let $G = (I, J, E)$ be a connected bipartite graph and let $\{V_A, V_B\}$ be a partition of the vertex set $I \cup J$ with $|V_A|, |V_B| \geq 2$. The *cut* determined by V_A is the set of edges $\delta(V_A) = \{(u, v) \in E : u \in V_A, v \notin V_A\}$. We say that the partition $\{V_A, V_B\}$ is a *split* if $\delta(V_A) = \{(i, j) : i \in I_B^1, j \in J_A^1\}$ holds for some $J_A^1 \subseteq V_A \cap J$ and $I_B^1 \subseteq V_B \cap I$, i.e. the edges of the cut determined by V_A form a biclique. The *split decomposition* of G at split $\{V_A, V_B\}$ is to delete edges $\delta(V_A)$ to obtain two bipartite graphs $G_A = (V_A, E_A)$ and $G_B = (V_B, E_B)$ and then add a new vertex i_A (j_B) to G_A (G_B) and connect it to all the vertices in J_A^1 (I_B^1). The reverse of a split decomposition is the *split-sum* of two bipartite graphs. If $G_A = (V_A, E_A)$ and $G_B = (V_B, E_B)$ are two bipartite graphs with $i_A \in V_A$ and $j_B \in V_B$, then the *split sum* of G_A and G_B is a bipartite graph $G = (V, E)$ with $V = (V_A \cup V_B) \setminus \{i_A, j_B\}$ and

$$E = \{(u, v) \in E_A \cup E_B : u, v \notin \{i_A, j_B\}\} \cup \{(i, j) : (i_A, j) \in E_A, (i, j_B) \in E_B\}.$$

Now we give the equivalent definition in terms of binary matrices. Let $\mathbf{A}$ and $\mathbf{B}$ be $m_A \times n_A$ and $m_B \times n_B$ binary matrices respectively. Let $I_A \cup \{i_A\}$ (where $i_A \notin I_A$)

and J_A denote the row and column index sets of $\mathbf{A}$ and let row i_A be a nonzero row of $\mathbf{A}$. Similarly, let I_B and $J_B \cup \{j_B\}$ (where $j_B \notin J_B$) denote the row and column index sets of $\mathbf{B}$ and let column j_B be a nonzero column of $\mathbf{B}$. Let $J_A^0 = \{j \in J_A : a_{i_A,j} = 0\}$ and $J_A^1 = \{j \in J_A : a_{i_A,j} = 1\}$ be a partition of J_A according to 0s and 1s in row i_A of $\mathbf{A}$. Let I_B^0 and I_B^1 be an analogously defined partition of I_B based on the 0s and 1s in column j_B of $\mathbf{B}$. Therefore, $\mathbf{A}$ and $\mathbf{B}$ have the form,

$$\mathbf{A} = \begin{array}{cc} & \begin{array}{cc} J_A^0 & J_A^1 \end{array} \\ \begin{array}{c} I_A \\ i_A \end{array} & \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \\ \mathbf{0}^\top & \mathbf{1}^\top \end{bmatrix} \end{array}, \qquad \mathbf{B} = \begin{array}{cc} & \begin{array}{cc} j_B & J_B \end{array} \\ \begin{array}{c} I_B^1 \\ I_B^0 \end{array} & \begin{bmatrix} \mathbf{1} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{B}_0 \end{bmatrix} \end{array}.$$

The Lubiw-sum operator or L-sum for short (simply called 'composition' when introduced in [77]), takes $\mathbf{A}$ and $\mathbf{B}$ as inputs and returns a binary matrix $\mathcal{L}^{(i_A,j_B)}(\mathbf{A},\mathbf{B}) \in \{0,1\}^{(m_A+m_B-1)\times(n_A+n_B-1)}$ which satisfies

$$\mathcal{L}^{(i_A,j_B)}(\mathbf{A},\mathbf{B})_{ij} = \begin{cases} a_{ij} & i \in I_A, j \in J_A, \\ b_{ij} & i \in I_B, j \in J_B, \\ 1 & i \in I_B^1, j \in J_A^1, \\ 0 & \text{otherwise.} \end{cases}$$

Visually, $\mathbf{A}$ and $\mathbf{B}$ are joined on a rectangle which is created from row i_A of $\mathbf{A}$ and column j_B of $\mathbf{B}$,

$$\mathcal{L}^{(i_A,j_B)}(\mathbf{A},\mathbf{B}) = \begin{array}{cc} & \begin{array}{ccc} J_A^0 & J_A^1 & J_B \end{array} \\ \begin{array}{c} I_A \\ I_B^1 \\ I_B^0 \end{array} & \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \\ & \mathbf{J} & \mathbf{B}_1 \\ & & \mathbf{B}_0 \end{bmatrix} \end{array}. \tag{3.3.1}$$

When the selected nonzero row i_A of $\mathbf{A}$ and the nonzero column j_B of $\mathbf{B}$ is clear from the context or arbitrary, we adopt the simpler notation $\mathcal{L}^{(i_A,j_B)}(\mathbf{A},\mathbf{B}) = \mathbf{A} \odot \mathbf{B}$. Lubiw [77, Lemma 3.2, Lemma 3.3] shows that the Boolean rank and isolation number of a matrix obtained by the L-sum operation satisfies the following lemmas, whose proofs we present in matrix form instead of their original proof in the bipartite setting.

Lemma 3.3.2. *[77, Lemma 3.2] The Boolean rank of* $\mathbf{A} \odot \mathbf{B}$ *satisfies*

$$\mathfrak{br}(\mathbf{A} \odot \mathbf{B}) = \min\left\{ \mathfrak{br}(\mathbf{A}) + \mathfrak{br}(\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_0 \end{bmatrix}),\ \ \mathfrak{br}(\begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \end{bmatrix}) + \mathfrak{br}(\mathbf{B}) \right\}.$$

Proof. ($\leq$) Any minimum rectangle cover of $\mathbf{A}$ can be expanded to a minimum rectangle cover of $\left[\begin{smallmatrix} \mathbf{A}_0 & \mathbf{A}_1 \\ & \mathbf{J} \end{smallmatrix}\right]$, hence a feasible rectangle cover of $\mathbf{A} \odot \mathbf{B}$ can be formed from the union of a minimum rectangle cover of $\mathbf{A}$ and of $\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]$. Similarly, a feasible rectangle cover of $\mathbf{A} \odot \mathbf{B}$ can be formed from the union of a minimum rectangle cover of $\mathbf{B}$ and of $[\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]$.

($\geq$) Let $\mathcal{R}$ be the set of rectangles in a minimum rectangle cover of $\mathbf{A} \odot \mathbf{B}$. Observe that 1s in $\mathbf{A}_0$ and $\mathbf{A}_1$ cannot be contained in a common rectangle with 1s in $\mathbf{B}_0$ and $\mathbf{B}_1$. Let $\mathcal{R}_A$ be the set of rectangles that cover a 1 from $[\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]$ and let $\mathcal{R}_B = \mathcal{R} \setminus \mathcal{R}_A$. Any rectangle in $\mathcal{R}_A$ can be transformed into a rectangle of $\mathbf{A}$ by compressing its rows corresponding to $\mathbf{J}$ into a single row. Similarly, any rectangle in $\mathcal{R}_B$ can be compressed into a rectangle of $\mathbf{B}$. Let $\mathcal{R}'_A$ be the rectangles of $\mathbf{A}$ obtained by compressing rectangles in $\mathcal{R}_A$ and let $\mathcal{R}'_B$ be the rectangles of $\mathbf{B}$ compressed from rectangles in $\mathcal{R}_B$. Then either (1.) $\mathcal{R}'_A$ is a proper cover of $\mathbf{A}$ or (2.) $\mathcal{R}'_B$ is a proper cover of $\mathbf{B}$, or both. In (1.) we have $|\mathcal{R}_A| = |\mathcal{R}'_A| \geq \mathfrak{br}(\mathbf{A})$ and $|\mathcal{R}_B| = |\mathcal{R}'_B| \geq \mathfrak{br}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$. While in (2.) we have $|\mathcal{R}_B| = |\mathcal{R}'_B| \geq \mathfrak{br}(\mathbf{B})$ and $|\mathcal{R}_A| = |\mathcal{R}'_A| \geq \mathfrak{br}([\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]$. □

Lemma 3.3.3. *[77, Lemma 3.3] The isolation number of* $\mathbf{A} \odot \mathbf{B}$ *satisfies*

$$\mathfrak{i}(\mathbf{A} \odot \mathbf{B}) = \max\left\{\mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1,\ \ \mathfrak{i}(\begin{bmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{bmatrix}) + \mathfrak{i}(\begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{bmatrix})\right\}.$$

Proof. ($\geq$) Since 1s in $\mathbf{A}_0$ and $\mathbf{A}_1$ cannot be contained in a common rectangle with 1s in $\mathbf{B}_0$ and $\mathbf{B}_1$, the union of a maximum isolated set of $[\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]$ and a maximum isolated set of $\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]$ is an isolated set of $\mathbf{A} \odot \mathbf{B}$, hence $\mathfrak{i}(\mathbf{A} \odot \mathbf{B}) \geq \mathfrak{i}([\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]) + \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$. Let S_A be a maximum isolated of $\mathbf{A}$, and S_B of $\mathbf{B}$.

(1.) If S_A contains a 1 from the last row of $\mathbf{A}$ with column index c, and S_B contains a 1 from the first column of $\mathbf{B}$ with row index r, then delete these two 1s and add $(\mathbf{A} \odot \mathbf{B})_{r,c}$ to obtain an isolated set for $\mathbf{A} \odot \mathbf{B}$ of size $\mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1$.

(2.) If S_A (S_B) contains a 1 from the last row of $\mathbf{A}$ (first column of $\mathbf{B}$) and S_B (S_A) does not contain a 1 from the first column of $\mathbf{B}$ (last row of $\mathbf{A}$), then remove that one from S_A (S_B) to obtain an isolated set for $\mathbf{A} \odot \mathbf{B}$ of size $\mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1$.

($\leq$) Let S be a maximum isolated set of $\mathbf{A} \odot \mathbf{B}$.

(1.) If S does not contain any 1s from block $\mathbf{J}$ then it can be split into two isolated sets, one of which is feasible for $[\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]$ and the other for $\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]$, so $\mathfrak{i}(\mathbf{A} \odot \mathbf{B}) \leq \mathfrak{i}([\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}]) + \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$.

(2.) Otherwise S contains a 1 from block $\mathbf{J}$. In this case $S \backslash \operatorname{supp}_1(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$ corresponds to a feasible isolated set of $\mathbf{A}$ and $S \backslash \operatorname{supp}_1([\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\end{smallmatrix}])$ corresponds to a feasible isolated set of $\mathbf{B}$ and they both contain the corresponding 1 in block $\mathbf{J}$, so $|S| \leq \mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1$. □

Using these lemmas, Lubiw proves that L-sum preserves firmness.

Theorem 3.3.4. *[77, Theorem 3.1] If* $\mathbf{A}$ *and* $\mathbf{B}$ *are firm then so is* $\mathbf{A} \odot \mathbf{B}$*. In particular, the L-sum operation preserves firmness.*

Proof. It suffices to show that $\mathfrak{i}(\mathbf{A} \odot \mathbf{B}) = \mathfrak{br}(\mathbf{A} \odot \mathbf{B})$ as any proper submatrix of $\mathbf{A} \odot \mathbf{B}$ is either a submatrix of $\mathbf{A}$, or $\mathbf{B}$ or of the form $\mathbf{A}' \odot \mathbf{B}'$, where $\mathbf{A}'$ and $\mathbf{B}'$ are submatrices of $\mathbf{A}$ and $\mathbf{B}$ respectively.

(1.) If $\mathfrak{i}(\mathbf{A}) > \mathfrak{i}([\,\mathbf{A}_0\ \mathbf{A}_1\,])$ and $\mathfrak{i}(\mathbf{B}) > \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$, then $\mathfrak{i}([\,\mathbf{A}_0\ \mathbf{A}_1\,]) = \mathfrak{i}(\mathbf{A}) - 1$ and $\mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]) = \mathfrak{i}(\mathbf{B}) - 1$, and by Lemma 3.3.3 we have $\mathfrak{i}(\mathbf{A} \odot \mathbf{B}) = \mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1$. Thus by Lemma 3.3.2 and firmness of $\mathbf{A}$ and $\mathbf{B}$ we have

$$\begin{aligned}\mathfrak{br}(\mathbf{A} \odot \mathbf{B}) &= \min\Big\{\mathfrak{br}(\mathbf{A}) + (\mathfrak{br}(\mathbf{B}) - 1),\ (\mathfrak{br}(\mathbf{A}) - 1) + \mathfrak{br}(\mathbf{B})\Big\}\\ &= \mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1 = \mathfrak{i}(\mathbf{A} \odot \mathbf{B}).\end{aligned}$$

(2.) If $\mathfrak{i}(\mathbf{A}) = \mathfrak{i}([\,\mathbf{A}_0\ \mathbf{A}_1\,])$, then $\mathfrak{br}(\mathbf{A}) = \mathfrak{br}([\,\mathbf{A}_0\ \mathbf{A}_1\,])$ holds as well by firmness. By Lemmas 3.3.2 and 3.3.3 we then have

$$\begin{aligned}\mathfrak{br}(\mathbf{A} \odot \mathbf{B}) &= \min\big\{\mathfrak{br}(\mathbf{A}) + \mathfrak{br}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]),\ \ \mathfrak{br}(\mathbf{A}) + \mathfrak{br}(\mathbf{B})\big\}\\ &= \mathfrak{br}(\mathbf{A}) + \mathfrak{br}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]),\\ \mathfrak{i}(\mathbf{A} \odot \mathbf{B}) &= \max\big\{\mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\mathbf{B}) - 1,\ \ \mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])\big\}\\ &= \mathfrak{i}(\mathbf{A}) + \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]),\end{aligned}$$

as $\mathfrak{br}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right]) \leq \mathfrak{br}(\mathbf{B})$ and $\mathfrak{i}(\mathbf{B}) - 1 \leq \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$ always hold. Therefore, $\mathfrak{br}(\mathbf{A} \odot \mathbf{B}) = \mathfrak{i}(\mathbf{A} \odot \mathbf{B})$.

(3.) If $\mathfrak{i}(\mathbf{B}) = \mathfrak{i}(\left[\begin{smallmatrix}\mathbf{B}_1\\ \mathbf{B}_0\end{smallmatrix}\right])$ then the same reasoning holds as in (2.). □

Lubiw goes one step further and proves that the L-sum operation also preserves superfirmness by showing that the L-sum of $\mathbf{A}$ and $\mathbf{B}$ can be seen as vertex replications and a clique sum applied to $\mathcal{G}(\mathbf{A})$ and $\mathcal{G}(\mathbf{B})$.

Theorem 3.3.5. *[77, Theorem 6.2] If* $\mathbf{A}$ *and* $\mathbf{B}$ *are superfirm then so is* $\mathbf{A} \odot \mathbf{B}$*. In particular, the L-sum operation preserves superfirmness.*

Proof. Let $\mathbf{A} \odot \mathbf{B} = \mathcal{L}^{(i_A, j_B)}(\mathbf{A}, \mathbf{B})$ for some row i_A of $\mathbf{A}$ and column j_B of $\mathbf{B}$. By our assumption the rectangle cover graphs $\mathcal{G}(\mathbf{A})$ and $\mathcal{G}(\mathbf{B})$ are perfect. By definition, the 1s of row i_A of $\mathbf{A}$ form a clique of size $|J_A^1|$ in $\mathcal{G}(\mathbf{A})$ and the 1s of column j_B form a clique of size $|I_B^1|$ in $\mathcal{G}(\mathbf{B})$. Observe, that $\mathcal{G}(\left[\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\\ & \mathbf{J}\end{smallmatrix}\right])$ contains a clique of size $|I_B^1| \cdot |J_A^1|$ which is obtained by replicating each vertex corresponding to a 1 in row i_A $|I_B^1|$ times. Similarly, $\mathcal{G}(\left[\begin{smallmatrix}\mathbf{J} & \mathbf{B}_1\\ & \mathbf{B}_0\end{smallmatrix}\right])$ contains a clique of size $|I_B^1| \cdot |J_A^1|$ which is obtained by replicating each vertex corresponding to a 1 in column j_B $|J_A^1|$ times. Finally, observe that $\mathcal{G}(\mathbf{A} \odot \mathbf{B})$ is the clique sum of $\mathcal{G}(\left[\begin{smallmatrix}\mathbf{A}_0 & \mathbf{A}_1\\ & \mathbf{J}\end{smallmatrix}\right])$ and $\mathcal{G}(\left[\begin{smallmatrix}\mathbf{J} & \mathbf{B}_1\\ & \mathbf{B}_0\end{smallmatrix}\right])$ on the clique of size $|I_B^1| \cdot |J_A^1|$ that is obtained by vertex replication in both graphs. Therefore, by the Clique-sum 3.1.10 and Replication Lemmas 3.1.6, $\mathcal{G}(\mathbf{A} \odot \mathbf{B})$ is perfect. □

Recall that to duplicate a row or column of a matrix is to add a new row or column to the matrix that is the copy of another row or column. We have already observed that the Boolean rank and isolation number are invariant under row and column duplication. Row and column duplication is a simple instance of the L-sum operation. For instance, to duplicate a row of $\mathbf{X}$ k times we take the L-sum of $\mathbf{X}$ with the all 1s column vector of size k. Since L-sum preserves firmness and superfirmness, row-column duplication preserves firmness and superfirmness.

Corollary 3.3.6. *[77] Let $\mathbf{X}$ be a binary matrix and let $\mathbf{X}'$ be obtained by duplicating some rows and/or columns of $\mathbf{X}$. If $\mathbf{X}$ is superfirm, then so is $\mathbf{X}'$; and if $\mathbf{X}$ is firm, then so is $\mathbf{X}'$. In particular, row and column duplication preserves firmness and superfirmness.*

A *chordal* graph is in which every cycle of four or more vertices has a chord. The proof of Theorem 3.3.5 may be further strengthened to show that the L-sum operation preserves chordality.

Corollary 3.3.7. *If $\mathcal{G}(\mathbf{A})$ and $\mathcal{G}(\mathbf{B})$ are chordal graphs then so is $\mathcal{G}(\mathbf{A} \odot \mathbf{B})$. In particular, the L-sum operation preserves chordality of the rectangle cover graph.*

Proof. By the proof of Theorem 3.3.5 it is sufficient to argue that vertex replication and the clique sum operation preserve chordality. Let G be a chordal graph and let G' be the graph obtained by replicating vertex v of G by new vertex v'. Since G and $G' \setminus \{v\}$ are both chordal, any cycle $\mathcal{C}$ in G' that is not an induced subgraph of G and $G' \setminus \{v\}$, must contain both v and v'. But then as v and v' are adjacent and have the same neighbours, if $|\mathcal{C}| \geq 4$ then a chord appears, hence G' is chordal too. Next, let G be the clique sum of two chordal graphs G_1 and G_2 on clique K. It is then easy to see that any cycle $\mathcal{C}$ that is not an induced subgraph of G_1 and G_2 would need to contain two vertices from K, which shows that $\mathcal{C}$ has a chord. Therefore, G is chordal. □

The L-sum operation may be extended to generalised binary matrices in a natural way. Let $\mathbf{A}, I_A, i_A, J_A^0, J_A^1$ and $\mathbf{B}, I_B^1, I_B^0, J_B, j_B$ be defined as previously and require that row i_A of $\mathbf{A}$ and column j_B of $\mathbf{B}$ have entries that are 1s as before. If row i_A and column j_B have some ?'s, then we can introduce two new blocks into $\mathbf{A}$ and $\mathbf{B}$,

$$\mathbf{A} = \begin{array}{c} \\ I_A \\ i_A \end{array} \begin{array}{c} \begin{array}{ccc} J_A^0 & J_A^? & J_A^1 \end{array} \\ \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_? & \mathbf{A}_1 \\ \mathbf{0}^\top & \mathbf{?}^\top & \mathbf{1}^\top \end{bmatrix} \end{array}, \qquad \mathbf{B} = \begin{array}{c} \\ I_B^1 \\ I_B^? \\ I_B^0 \end{array} \begin{array}{c} \begin{array}{cc} j_B & J_B \end{array} \\ \begin{bmatrix} \mathbf{1} & \mathbf{B}_1 \\ \mathbf{?} & \mathbf{B}_? \\ \mathbf{0} & \mathbf{B}_0 \end{bmatrix} \end{array}.$$

Then the L-sum of $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\mathcal{L}^{(i_A,j_B)}(\mathbf{A},\mathbf{B}) = \begin{array}{c|cccc|} & J_A^0 & J_A^? & J_A^1 & J_B \\ \hline I_A & \mathbf{A}_0 & \mathbf{A}_? & \mathbf{A}_1 & \\ I_B^1 & & ? & \mathbf{J} & \mathbf{B}_1 \\ I_B^? & & ? & ? & \mathbf{B}_? \\ I_B^0 & & & & \mathbf{B}_0 \\ \hline \end{array}.$$

One can double check that the proofs of Lemmas 3.3.2, 3.3.3 and Theorems 3.3.4, 3.3.5 hold for L-sums of generalised binary matrices as well. Therefore, L-sum preserves firmness of generalised binary matrices as well.

L-decomposable matrices. Reversing the L-sum operation is equivalent to determining whether a matrix $\mathbf{X}$ arises from smaller matrices by the L-sum operation. If $\mathbf{X} = \mathbf{A} \odot \mathbf{B}$ for some matrices $\mathbf{A}, \mathbf{B}$ then we say that $\mathbf{X}$ is *L-decomposable.* By Theorems 3.3.4 if $\mathbf{X}$ is a matrix that is L-decomposable to matrices that are firm, then $\mathbf{X}$ is firm.

A standard binary matrix $\mathbf{X}$ is L-decomposable if and only if the bipartite representation $\mathcal{B}(\mathbf{X})$ is split decomposable. In [25], Cunningham and Edmonds show that in polynomial time, every connected graph can be uniquely split decomposed into a minimum number of components, where each component is a prime graph (a graph that cannot be further split decomposed), a star or a clique. Stars and cliques are not prime graphs, as stars can be split decomposed into smaller stars, while cliques can be split decomposed into smaller cliques. However, to get a minimum number of components in a split decomposition, one should not split decompose stars and cliques.

A class of binary matrices is called *hereditary* if it is closed under taking submatrices. Lubiw [77, Section 4] uses Cunningham's split decomposition algorithm to show that there is a polynomial time algorithm to compute a minimum rectangle cover and a maximum isolated set for any class of binary matrices that is a hereditary class *and* for which there exists a polynomial time algorithm to compute a minimum rectangle cover and a maximum isolated set on the prime components that arise in its split decomposition.

A graph G is *distance hereditary* if every cycle of length 5 or more has at least two crossing chords, where two chords (a, c) and (b, d) are *crossing* if the four vertices a, b, c, d lie in this order on the cycle. A graph is distance hereditary if and only if its split decomposition consists of only stars and cliques [52, 40]. Distance hereditary graphs have a forbidden subgraph characterisation, which says that a graph G is distance hereditary if and only if it is domino, gem and house-free and does not have any holes of length 5 or more [4], [12, Theorem 10.1.1]. A *gem* is a cycle of length 5 with two non-crossing chords and a *house* is a cycle of length 5 with only one chord. It follows that a graph is *bipartite distance hereditary* if and only if it is domino-free and chordal bipartite. Furthermore, the result of [52] on the split decomposition of distance hereditary graphs implies that a bipartite graph is distance hereditary if and only if its split decomposition consists of only stars.

Recall that the domino graph's biadjacency matrix is $\mathbf{D}_3$. In the binary matrix setting, bipartite distance hereditary graphs are the bipartite representation of totally balanced matrices that have no $\mathbf{D}_3$ submatrix. As stars are just the bipartite representations of row and column binary vectors, we get the following theorem.

Theorem 3.3.8 (Corollary of [4] and [52]). *$\mathbf{X} \in \{0,1\}^{m\times n}$ can be L-decomposed into row and column vectors if and only if $\mathbf{X}$ has no $\mathbf{D}_3$ and no $\mathbf{C}_k$ submatrices for any $k \geq 3$.*

The rectangle cover graph of row and column vectors are just cliques. By Theorem 3.3.5 the L-sum operation preserves superfirmness therefore $\mathbf{D}_3$-free totally balanced matrices are another class of superfirm matrices. One can even say a stronger argument by using the result that L-sum preserves chordality of the rectangle cover graph.

Theorem 3.3.9. *Let $\mathbf{X} \in \{0,1\}^{m\times n}$. $\mathcal{G}(\mathbf{X})$ is chordal if and only if $\mathbf{X}$ has no $\mathbf{D}_3$ and no $\mathbf{C}_k$ submatrices for any $k \geq 3$.*

Proof. It is clear that the rectangle cover graphs of $\mathbf{D}_3$ and $\mathbf{C}_k$ ($k \geq 3$) are not chordal as $\mathcal{G}(\mathbf{C}_k)$ is a $2k$-hole and $\mathcal{G}(\mathbf{D}_3)$ contains a 4-hole as shown in Figure 3.6 below.

Now let $\mathbf{X}$ be a binary matrix with no $\mathbf{D}_3$ and no $\mathbf{C}_k$ ($k \geq 3$) submatrix. Then $\mathbf{X}$ is L-decomposable into row and column vectors by Theorem 3.3.8 and equivalently $\mathbf{X}$ may be built up from binary row and column vectors by the L-sum operation. Since the rectangle cover graphs of row and column vectors are just cliques, and the L-sum operation preserves chordality by Corollary 3.3.7, $\mathcal{G}(\mathbf{X})$ is chordal. □

As linear matrices are another class of superfirm matrices, Lubiw had the very interesting idea to analyse the closure under the L-sum operation of linear matrices in terms of forbidden submatrices. We say a binary matrix is *L-decomposable into linear matrices* if it is linear or can be obtained by a series of L-sums applied to

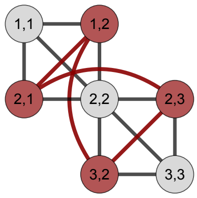
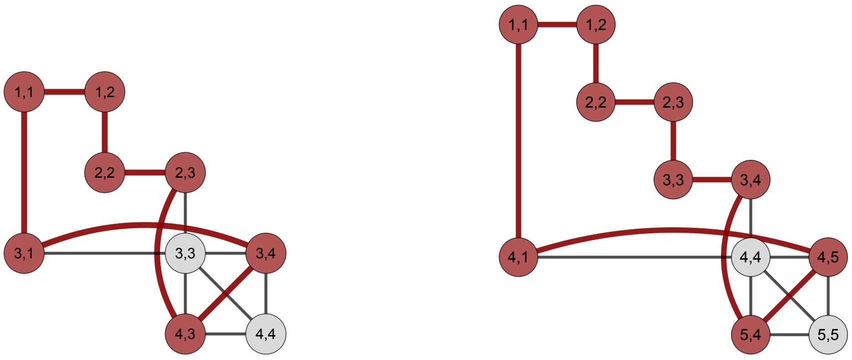


Figure 3.6: $\mathcal{G}(\mathbf{D}_3)$ with its 4-hole highlighted

linear matrices [77, pg.108]. Observe that these matrices then are superfirm, as L-sum preserves superfirmness and linear matrices are superfirm. For $n \geq 4$, let $\mathbf{M}_n$ be the $n \times n$ matrix,

$$\mathbf{M}_n := \begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ 1 & & & & 1 & 1 \\ & & & & 1 & 1 \end{bmatrix}. \tag{3.3.2}$$

The rectangle cover graph of $\mathbf{M}_n$ contains an odd hole of size $2n-1$, $\mathcal{G}(\mathbf{M}_n)$ with the $2n-1$-hole highlighted can be seen in Figure 3.7 for $n = 4, 5$.



(a) $\mathcal{G}(\mathbf{M}_4)$

(b) $\mathcal{G}(\mathbf{M}_5)$

Figure 3.7: The rectangle cover graph of $\mathbf{M}_4$ and $\mathbf{M}_5$ with their odd hole highlighted

The following theorem of Lubiw, provides a forbidden submatrix characterisation of matrices that are L-decomposable into linear matrices.

Theorem 3.3.10. *[77, Lemma 5.3] $\mathbf{X} \in \{0,1\}^{m \times n}$ is L-decomposable into linear matrices if and only if it does not have $\mathbf{D}_3$ and $\mathbf{M}_k$ submatrices for any $k \geq 4$.*

Lubiw notes that strongly balanced matrices are L-decomposable into linear pieces. A binary matrix $\mathbf{X}$ is said to be *balanced* if it has no $\mathbf{C}_n$ submatrix for any *odd* $n \geq 3$, and it is said to be *strongly balanced* if it is balanced and any matrix obtained from it by changing a single 1 to a 0 is also balanced. Matrices $\mathbf{M}_n$ for even n are not balanced, and for odd n they are balanced but not strongly balanced. In addition, $\mathbf{D}_3$ is also not strongly balanced. Hence, strongly balanced matrices cannot have a $\mathbf{D}_3$ or $\mathbf{M}_n$ submatrix and they are L-decomposable to linear pieces.

Furthermore, note that as $\mathbf{M}_k$ is not superfirm for any $k \geq 4$ and clearly has no $\mathbf{D}_3$ submatrix, the following statement is true for $\mathbf{D}_3$-free matrices.

Corollary 3.3.11. *Let $\mathbf{X} \in \{0,1\}^{m \times n}$ have no $\mathbf{D}_3$ submatrix. Then $\mathbf{X}$ is superfirm if and only if it has no $\mathbf{M}_k$ submatrix for any $k \geq 4$.*

The above corollary implies that $\mathbf{D}_3$-free superfirm matrices are exactly the matrices that can be L-decomposed into linear pieces. Clearly $\mathbf{D}_3$ is superfirm itself, hence there is more work to do to get a complete characterisation of superfirm matrices in terms of forbidden submatrices.

3.3.3 $\mathbf{D}_3$-free matrices

Forbidding $\mathbf{D}_3$ leads to some beautiful results most of which come from a 1998 paper by Amilhastre et al. [2]. In their paper, it is proved that the Boolean rank of $\mathbf{D}_3$-free binary matrices can be computed in polynomial time. It is also shown that the Boolean rank of $\mathbf{D}_3$-free matrices equals the minimum number of rectangles needed to partition the 1s of the matrix, the rectangle partition number. Interestingly, the isolation number of $\mathbf{D}_3$-free matrices is not mentioned to be equal to the Boolean rank. Their constructions however, along with a brief complementary result that we prove in this section, allow us to deduce that the isolation number of $\mathbf{D}_3$-free matrices equals the Boolean rank, and by this $\mathbf{D}_3$-free matrices form another class of firm matrices. Recall that matrices $\mathbf{M}_n$ have no $\mathbf{D}_3$ submatrix but contain an odd hole in their rectangle cover graph as shown in Figure 3.7. Hence $\mathbf{D}_3$-free matrices give a class of firm matrices that is not a subset of superfirm matrices.

Let us briefly describe the main ideas in [2]. Recall from Section 1.1.1 that a binary matrix is a row-clutter matrix if its rows are element-wise incomparable. We say that a matrix is a *row-column-clutter* matrix if it is row-clutter and column-clutter. The main results of [2] can be divided into two parts. In the first part, it is proved that any $\mathbf{D}_3$-free binary matrix $\mathbf{X}$ that is not a row-column-clutter matrix can be reduced in polynomial time into a row-column-clutter matrix $\mathbf{X}'$ that is also $\mathbf{D}_3$-free and the Boolean rank of $\mathbf{X}$ equals the Boolean rank of $\mathbf{X}'$. We will show that this reduction also preserves the isolation number of $\mathbf{X}$. In the second part, given a $\mathbf{D}_3$-free row-column-clutter matrix $\mathbf{X}$ a graph $\mathcal{H}(\mathbf{X})$ is defined whose vertex set is an extended set of maximal rectangles of $\mathbf{X}$. Then it is shown that $\mathcal{C}$ is a minimum rectangle cover of $\mathbf{X}$ if and only if it is a minimum vertex-cut of $\mathcal{H}(\mathbf{X})$.

Reduction procedure. Let us first illustrate the procedure by which a $\mathbf{D}_3$-free matrix $\mathbf{X}$ can be reduced to a row-column-clutter matrix. For a row ℓ of $\mathbf{X}$, let $succ(\ell)$ contain the indices of all the rows different from row ℓ that are element-wise greater than or equal to row ℓ,

$$succ(\ell) := \{i \neq \ell : \mathbf{X}_{\ell,:} \leq \mathbf{X}_{i,:}\}.$$

For a column k, define $succ(k)$ analogously. Observe that if row ℓ is a row that is strictly maximal among the other rows with respect to element-wise $\leq$, so $succ(\ell) = \emptyset$, then row ℓ forms a maximal rectangle.

For a non-zero row ℓ of $\mathbf{X}$ with $succ(\ell) \neq \emptyset$, by *reducing* $\mathbf{X}$ *on* ℓ [2, Definition 4.2] we obtain another matrix $\mathbf{X}'$ whose rows satisfy

$$\mathbf{X}'_{i,:} = \begin{cases} \mathbf{X}_{i,:} - \mathbf{X}_{\ell,:} & \text{if } i \neq \ell \text{ and } \mathbf{X}_{\ell,:} \leq \mathbf{X}_{i,:}, \\ \mathbf{X}_{i,:} & \text{otherwise.} \end{cases}$$

If after a row reduction $\mathbf{X}'$ has some 0 rows, we delete those rows. Reduction on a column k is defined analogously, with possible 0 columns deleted. Amilhastre et al. prove that if $\mathbf{X}$ is a $\mathbf{D}_3$-free binary matrix, then

- the matrix obtained by reducing $\mathbf{X}$ on a row is $\mathbf{D}_3$-free too [2, Property 4.3],
- the Boolean rank of the matrix $\mathbf{X}'$ obtained by reducing $\mathbf{X}$ on a row equals the Boolean rank of $\mathbf{X}$ and a minimum rectangle cover of $\mathbf{X}'$ can be extended back to give a minimum rectangle cover of $\mathbf{X}$ [2, Property 4.2].

In addition, Amilhastre et al. prove that performing a row reduction does not create new column reductions [2, Property 4.4]. Hence one can first do all row reductions and then do all column reductions. They show that for a matrix $\mathbf{X} \in \{0,1\}^{m \times n}$ with $|\operatorname{supp}_1(\mathbf{X})| = q$, one can obtain a matrix that cannot be further row-reduced in $\mathcal{O}(mq)$ time [2, Lemma 4.2]. Therefore, a matrix that cannot be further row nor column reduced can be obtained in $\mathcal{O}((m+n)q)$ time. Since a matrix that cannot be further reduced has $succ(i) = succ(j) = \emptyset$ for all rows i and columns j, it is a row-column-clutter matrix. Therefore, if there is a polynomial time algorithm to compute a minimum rectangle cover of row-column-clutter $\mathbf{D}_3$-free matrices, then there is also a polynomial time algorithm to compute a minimum rectangle cover of arbitrary $\mathbf{D}_3$-free binary matrices.

Does the reduction operation preserve isolation number of a $\mathbf{D}_3$-free binary matrix? While this question is not treated in [2], in the next theorem we show that this is indeed the case.

Theorem 3.3.12. *If* $\mathbf{X}$ *is a* $\mathbf{D}_3$*-free binary matrix, then the isolation number of the matrix obtained by reducing* $\mathbf{X}$ *on a row equals the isolation number of* $\mathbf{X}$*.*

Proof. Let row ℓ be a non-zero row of $\mathbf{X}$ with $succ(\ell) \neq \emptyset$ and let $\mathbf{X}'$ be obtained by reducing $\mathbf{X}$ on row ℓ.

We show first that $\mathfrak{i}(\mathbf{X}') \leq \mathfrak{i}(\mathbf{X})$. Let $S' \subseteq \operatorname{supp}_1(\mathbf{X}')$ be a maximum isolated set of $\mathbf{X}'$. Then S' is also a subset of $\operatorname{supp}_1(\mathbf{X})$. Suppose that $(i_1, j_1), (i_2, j_2) \in S'$ are not isolated in $\mathbf{X}$, so $(i_1, j_2), (i_2, j_1) \in \operatorname{supp}_1(\mathbf{X})$. Then the 0 that makes them isolated in $\mathbf{X}'$ is created by the reduction on row ℓ and we must have either $i_1 \in succ(\ell)$ or $i_2 \in succ(\ell)$ but not both (having both $i_1, i_2 \in succ(\ell)$ would mean $(\ell, j_1), (\ell, j_2) \in \operatorname{supp}_1(\mathbf{X})$ and none of $(i_1, j_1), (i_2, j_2)$ could be 1s in $\mathbf{X}'$). Without loss of generality assume that $i_2 \in succ(\ell)$ and $i_1 \notin succ(\ell)$. Then (i_2, j_1) was turned into a 0 by the reduction on row ℓ so $(\ell, j_1) \in \operatorname{supp}_1(\mathbf{X})$. Similarly, (i_2, j_2) stayed a 1 after the reduction on row ℓ so $(\ell, j_2) \in \operatorname{supp}_0(\mathbf{X})$. As $i_1 \notin succ(\ell)$ and $\ell \notin succ(i_1)$ there exists a k for which $(i_1, k) \in \operatorname{supp}_0(\mathbf{X})$ and $(\ell, k) \in \operatorname{supp}_1(\mathbf{X})$. But then as $i_2 \in succ(\ell)$, we also have $(i_2, k) \in \operatorname{supp}_1(\mathbf{X})$. The submatrix of $\mathbf{X}$ formed by rows $\{i_1, i_2, \ell\}$ and columns $\{j_2, j_1, k\}$ then forms a $\mathbf{D}_3$ submatrix, a contradiction,

$$\begin{array}{c|ccc} & j_2 & j_1 & k \\ \hline i_1 & 1 & 1 & 0 \\ i_2 & 1 & 1 & 1 \\ \ell & 0 & 1 & 1 \end{array}.$$

Therefore, S' must be a feasible isolated set of $\mathbf{X}$, and $|S'| \leq \mathfrak{i}(\mathbf{X})$.

Next, we show that $\mathfrak{i}(\mathbf{X}) \leq \mathfrak{i}(\mathbf{X}')$. Let S be a maximum isolated set of $\mathbf{X}$. By reducing on a row, new 1s are never created hence if all 1s in S remain 1s in $\mathbf{X}'$ then they are isolated and we are fine. So suppose that some 1s from S are turned to 0s by the reduction on row ℓ. Suppose there exist two distinct such 1s $(i_1, j_1), (i_2, j_2) \in S$. Then their rows i_1, i_2 are in $succ(\ell)$ and $(\ell, j_1) \in \operatorname{supp}_1(\mathbf{X})$ and $(\ell, j_2) \in \operatorname{supp}_1(\mathbf{X})$. But then by $i_1, i_2 \in succ(\ell)$, we also have $(i_1, j_2), (i_2, j_1) \in \operatorname{supp}_1(\mathbf{X})$ which contradicts their isolation, so there is at most one element from S that may be turned to a 0 in $\mathbf{X}'$. Let $(i, j) \in S$ be this unique element. Then again we have $i \in succ(\ell)$, so $(\ell, j) \in \operatorname{supp}_1(\mathbf{X})$. Let $S' = (S \setminus \{(i, j)\}) \cup \{(\ell, j)\}$ which is now a subset of $\operatorname{supp}_1(\mathbf{X}')$. We argue that S' is still isolated in $\mathbf{X}$. Suppose it is not and there is $(i_1, j_1) \in S \setminus \{(i, j)\}$

which is in a common rectangle in $\mathbf{X}$ with (ℓ, j). Then $(i_1, j), (\ell, j_1) \in \operatorname{supp}_1(\mathbf{X})$. But then since $i \in \mathit{succ}(\ell)$, $(i, j_1) \in \operatorname{supp}_1(\mathbf{X})$ which shows that (i, j) and (i_1, j_1) are not isolated, a contradiction. Therefore, S' is another maximum isolated set of $\mathbf{X}$ and it is also a feasible isolated set of $\mathbf{X}'$, so $|S| \leq \mathfrak{i}(\mathbf{X}')$. □

By this complementary result, similarly to a minimum rectangle cover, if there is a polynomial time algorithm to compute a maximum isolated set of row-column-clutter $\mathbf{D}_3$-free matrices, there is also a polynomial time algorithm to compute a maximum isolated set of arbitrary $\mathbf{D}_3$-free binary matrices.

Row-column-clutter $\mathbf{D}_3$-free matrices. What kind of overlap is possible between the rectangles of $\mathbf{D}_3$-free matrices? In general, for any binary matrix, if $I_1 \times J_1$ and $I_2 \times J_2$ are two distinct maximal rectangles, then we can have $I_1 \subset I_2$ if and only if $J_2 \subset J_1$ [2, Property 3.1], as otherwise there is a contradiction to the maximality of one of the rectangles. The following theorem shows how forbidding $\mathbf{D}_3$ submatrices influences the intersection maximal rectangles can have.

Theorem 3.3.13. *[2, Theorem 3.1] A binary matrix $\mathbf{X}$ is $\mathbf{D}_3$-free if and only if for any distinct maximal rectangles $I_1 \times J_1, I_2 \times J_2$ such that $(I_1 \times J_1) \cap (I_2 \times J_2) \neq \emptyset$, one of the statements is true: (i) $I_1 \subset I_2$ and $J_2 \subset J_1$, (ii) $I_2 \subset I_1$ and $J_1 \subset J_2$.*

Recall that $\mathcal{R}_{\max}(\mathbf{X})$ denotes the set of maximal rectangles of $\mathbf{X}$. For an $m \times n$ matrix $\mathbf{X}$, let $\top = \emptyset \times [n]$ and $\bot = [m] \times \emptyset$ be two dummy maximal rectangles and define the set of extended maximal rectangles as $\mathcal{R}^*_{\max}(\mathbf{X}) := \mathcal{R}_{\max}(\mathbf{X}) \cup \{\top, \bot\}$. Let $\mathcal{H}(\mathbf{X})$ be a simple undirected graph[1] on vertex set $\mathcal{R}^*_{\max}(\mathbf{X})$ where two maximal rectangles $I_1 \times J_1, I_2 \times J_2 \in \mathcal{R}^*_{\max}$ are adjacent if

(i) $I_1 \subseteq I_2$ and $J_2 \subseteq J_1$,

(ii) and there is no other rectangle $I \times J \in \mathcal{R}^*_{\max}$ such that $I_1 \subsetneq I \subsetneq I_2$ and $J_2 \subsetneq J \subsetneq J_1$.

Furthermore, for all $(i, j) \in \operatorname{supp}_1(\mathbf{X})$ we define the subset of maximal rectangles that cover (i, j),

$$\mathcal{R}_{(i,j)}(\mathbf{X}) := \{I \times J \in \mathcal{R}_{\max}(\mathbf{X}) : (i, j) \in I \times J\}.$$

Let us quickly review some graph terminology before we state the main theorem of Amilhastre et al. Let $G = (V, E)$ be a graph with $s, t \in V$ being two non-adjacent

[1] In the original setting of [2], $\mathcal{H}$ is defined to be the graph whose natural drawing is the Hasse diagram of the partially ordered set on $\mathcal{R}^*_{\max}$ under relation (i), but we prefer to directly go to the graph definitions for the sake of succinctness.

vertices of G. An *s, t-path* in G is a path which connects s and t. An *s, t-vertex-cut* of G is a subset of $V \setminus \{s, t\}$ whose removal disconnects s and t. Amilhastre et al. prove the following about $\mathcal{H}(\mathbf{X})$.

Theorem 3.3.14. *[2, Theorem and Corollary 5.1] Let $\mathbf{X}$ be a row-column-clutter $\mathbf{D}_3$-free binary matrix. Then the following two statements hold.*

(1) $P \subseteq \mathcal{R}^*_{\max}(\mathbf{X})$ is an $\top, \bot$-path in $\mathcal{H}(\mathbf{X})$ if and only if $P = \mathcal{R}_{(i,j)}(\mathbf{X}) \cup \{\top, \bot\}$ for some $(i, j) \in \operatorname{supp}_1(\mathbf{X})$.

(2) $\mathcal{C} \subseteq \mathcal{R}_{\max}(\mathbf{X})$ is a feasible rectangle cover of $\mathbf{X}$ if and only if $\mathcal{C}$ is a $\top, \bot$-vertex-cut of $\mathcal{H}(\mathbf{X})$.

Therefore, a minimum rectangle cover of a row-column clutter $\mathbf{D}_3$-free matrix is equivalent to a minimum $\top, \bot$-vertex-cut of $\mathcal{H}(\mathbf{X})$. Amilhastre et al. [2] argue that computing a minimum $\top, \bot$-vertex-cut can be done in polynomial time by network flow techniques. However, for us their theorem shows an even more interesting result than just the polynomial computability of a minimum rectangle cover.

Observe that S is an isolated set of $\mathbf{X}$ if and only if for any distinct $(i_1, j_1), (i_2, j_2) \in S$ we have $\mathcal{R}_{(i_1,j_1)}(\mathbf{X}) \cap \mathcal{R}_{(i_2,j_2)}(\mathbf{X}) = \emptyset$. Therefore, $S \subseteq \operatorname{supp}_1(\mathbf{X})$ is an isolated set of $\mathbf{X}$ if and only if the $\top, \bot$-paths $\mathcal{R}_{(i,j)}(\mathbf{X}) \cup \{\top, \bot\}$ corresponding to $(i, j) \in S$ are pairwise internally vertex disjoint in $\mathcal{H}(\mathbf{X})$.

Menger's theorem states that for any graph G the size of a minimum s, t-vertex-cut of G is equal to the maximum number of pairwise internally vertex-disjoint s, t-paths in G. Therefore, Menger's theorem implies that the maximum number of pairwise internally vertex disjoint $\top, \bot$-paths of $\mathcal{H}(\mathbf{X})$ is equal to the cardinality of a minimum $\top, \bot$-vertex-cut of $\mathcal{H}(\mathbf{X})$. As pairwise internally vertex disjoint $\top, \bot$-paths of $\mathcal{H}(\mathbf{X})$ are in direct correspondence with isolated sets of $\mathbf{X}$ and $\top, \bot$-vertex-cuts of $\mathcal{H}(\mathbf{X})$ with rectangle covers of $\mathbf{X}$, for any row-column clutter $\mathbf{D}_3$-free matrix we have $\mathfrak{i}(\mathbf{X}) = \mathfrak{br}(\mathbf{X})$. In addition, as any $\mathbf{D}_3$-free matrix can be reduced to a row-column clutter $\mathbf{D}_3$-free matrix and the reduction preserves $\mathfrak{i}(\mathbf{X})$ and $\mathfrak{br}(\mathbf{X})$, we have $\mathfrak{i}(\mathbf{X}) = \mathfrak{br}(\mathbf{X})$ for any $\mathbf{D}_3$-free matrix. Furthermore, since any submatrix of a $\mathbf{D}_3$-free matrix is also $\mathbf{D}_3$-free we arrive at the main theorem of this section.

Theorem 3.3.15. *If a binary matrix has no $\mathbf{D}_3$ submatrix then it is firm.*

We mention that Amilhastre et al's construction along with Menger's theorem, in addition to providing a polynomial time algorithm to compute a minimum rectangle cover of $\mathbf{D}_3$-free matrices, also provides a polynomial time algorithm to find a maximum isolated set of $\mathbf{D}_3$-free matrices.

Could there be a proof which circumvents the use of graph $\mathcal{H}(\mathbf{X})$? We suspect this to be the case, and conjecture that it should be possible to show that every row-column-clutter $\mathbf{D}_3$-free matrix has full isolation number.

Conjecture 3.3.16. *If* $\mathbf{X} \in \{0,1\}^{m\times n}$ *is a* $\mathbf{D}_3$*-free matrix that is row-column-clutter, then* $\mathfrak{i}(\mathbf{X}) = \min\{m, n\}$.

This conjecture together with Amilhastre et al's reduction procedure would imply firmness of $\mathbf{D}_3$-free matrices. However, note that row-column-clutter $\mathbf{D}_3$-free matrices are not superfirm, because while $\mathbf{M}_n$ is not row-column-clutter, the below $\mathbf{D}_3$-free matrix with two $\mathbf{M}_3$ submatrices is,

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

3.3.4 Interval matrices

Recall that a binary matrix is an interval matrix if there is an ordering of its columns making the 1s in each row consecutive. Interval matrices form a strict subset of totally balanced matrices and can be recognised in polynomial time [41, Chapter 8.3]. There is also a forbidden submatrix characterisation of interval matrices [104]. A powerful theorem of Győri [50] shows that interval matrices are firm.

The original proof of Győri's Theorem is non-algorithmic and remarkably difficult and long. There appeared several simplifications of it over the years, one of which is an algorithmic version by Franzblau et al. [37], that also provides a polynomial time algorithm to compute a minimum rectangle cover and maximum isolated set for interval matrices. Another reproof uses the concept of bisupermodularity [99, 60.3d] and another one uses partially ordered sets [36].

Győri's Theorem is also special because interval matrices are firm matrices that are not necessarily superfirm. Recall that $\mathbf{D}_4$ is a row-column interval matrix which has a 5-hole in its rectangle cover graph as shown in Figure 3.2, hence it is not superfirm.

Actually, Győri proves an even stronger statement than the firmness of interval matrices. He proves that for any interval matrix the cardinality of a maximum isolated sequence [78] is equal to the cardinality of a minimum rectangle cover. An *isolated sequence* of $\mathbf{X}$ is an isolated set $S \subseteq \operatorname{supp}_1(\mathbf{X})$ which satisfies the extra property that it can be ordered as $(i_1, j_1), (i_2, j_2), \ldots, (i_{|S|}, j_{|S|})$ such that the submatrix of $\mathbf{X}$ indexed and ordered by $\{i_1, i_2, \ldots, i_{|S|}\} \times \{j_1, j_2, \ldots, j_{|S|}\}$ is an upper triangular matrix,

$$\begin{array}{cc} & \begin{array}{ccccc} j_1 & j_2 & & & j_{|S|} \end{array} \\ \begin{array}{c} i_1 \\ i_2 \\ \\ \\ i_{|S|} \end{array} & \begin{bmatrix} 1 & * & * & \ldots & * \\ & 1 & * & \ldots & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & 1 & * \\ & & & & 1 \end{bmatrix}. \end{array}$$

On a side note we mention that Lubiw later on shows that a matrix $\mathbf{X}$ is totally balanced if and only if every isolated set of $\mathbf{X}$ is an isolated sequence [78, Theorem 5.1]. This result also shows that for totally balanced matrices we have $\mathbf{i}(\mathbf{X})$ less than or equal to the real rank of $\mathbf{X}$ as the triangular submatrix corresponding to the isolated sequence has full real rank.

We will illustrate the firmness of interval matrices via highlights from the algorithmic proof of Franzblau et al. [37]. The key element of their algorithm is a reduction procedure on an interval matrix $\mathbf{X}$ which replaces $\mathbf{X}$ by another interval matrix that has one less row but the same cardinality of maximum isolated sequence and the same cardinality of minimum rectangle cover as $\mathbf{X}$.

Let us introduce some new notation specific to the interval structure of the matrices considered. Let $[k, \ell] := \{k, k+1, \ldots, \ell-1, \ell\}$ for positive integers $k \leq \ell$ be called a *unit interval* with left endpoint k and right endpoint ℓ. For a positive integer n, let $\mathcal{I}(n)$ denote the set of all unit intervals that are a subset of $[n]$,

$$\mathcal{I}(n) = \{[k, \ell] : 1 \leq k \leq \ell \leq n\}.$$

Given an interval matrix $\mathbf{X} \in \{0,1\}^{m \times n}$, for each row $i \in [m]$ of $\mathbf{X}$ we can associate a unit interval in $\mathcal{I}(n)$,

$$\mathbf{X}_{i,:} = \sum_{j=k}^{\ell} \boldsymbol{e}_j^\top \in \{0,1\}^{1 \times n} \iff X_i = [k, \ell] \in \mathcal{I}(n),$$

where $\boldsymbol{e}_j$ is the j-th standard unit vector. By slight abuse of notation let $X = \{X_1, \ldots, X_m\} \subseteq \mathcal{I}(n)$, the "unit interval form" of $\mathbf{X}$.

If $\mathbf{X}$ is an interval matrix, we may assume that in any feasible factorisation $\mathbf{A} \circ \mathbf{B}$ of $\mathbf{X}$, $\mathbf{B}$ is an interval matrix, since each row of $\mathbf{X}$ satisfies $\mathbf{X}_{i,:} = \mathbf{A}_{i,:} \circ \mathbf{B}$. Hence every feasible factorisation of $\mathbf{X}$ may be associated with a set of unit intervals $B \subset \mathcal{I}(n)$, such that every unit interval in X can be expressed as the union of some unit intervals in B. We call B a *generating set* of X. Note that X itself is a generating set (but most likely not of minimum cardinality) corresponding to the factorisation $\mathbf{I}_m \circ \mathbf{X}$.

An isolated sequence S of $\mathbf{X}$ can also be seen as a set of $|S|$ unit intervals $\{J_1, \ldots, J_{|S|}\} \subseteq X$, such that each J_t contains an element $j \in J_t$ so that j is not an element of $J_1, \ldots, J_{t-1}$. We say in this case that the set of $|S|$ unit intervals $J_1, \ldots, J_{|S|}$ is an isolated sequence.

A subset $X_I := \{X_i \in X : i \in I\}$ of X is called *dependent* [37] if $\cup_{i \in I} X_i$ is a unit interval $[k, \ell] \in \mathcal{I}(n)$ and every $j \in [k, \ell]$ is contained in at least two members of X_I. Franzblau et al. prove the following.

Lemma 3.3.17. *[37, Lemma 3.3] $S \subset \mathcal{I}(n)$ is an isolated sequence if and only if S has no dependent subset.*

A unit interval $[k, \ell] \in \mathcal{I}(n)$ is *dependent in $X \subset \mathcal{I}(n)$* if $\{X_i \in X : X_i \subseteq [k, \ell]\}$ is a dependent subset of X. Furthermore a unit interval $[k, \ell]$ that is dependent in X is said to be *minimally dependent in X* if no proper subinterval $[k', \ell'] \subsetneq [k, \ell]$ is dependent in X.

A unit interval in $T \subseteq \mathcal{I}(n)$ is called *maximal in T* if it is not contained in any other unit interval of T.

The core of Franzblau et al.'s algorithm is a reduction procedure which reduces X on a dependent unit interval $[\ell, k]$ in X. Let $J_1, J_2, \ldots, J_t$ be the set of maximal unit intervals in the dependent subset $\{X_i \in X : X_i \subseteq [k, \ell]\}$ corresponding to the dependent interval $[\ell, k]$, ordered by left endpoint (and by also right endpoint then). Then a reduction step on X using $[k, \ell]$ outputs X',

$$X' = \begin{cases} X \setminus \{J_1\} & \text{if } t = 1, \\ X \cup \{J_1 \cap J_2, J_2 \cap J_3, \ldots, J_{t-1} \cap J_t\} \setminus \{J_1, J_2, \ldots, J_t\} & \text{if } t > 1. \end{cases}$$

Since $[\ell, k]$ is dependent in X, every $j \in [\ell, k]$ is covered by at least two unit intervals of X, so X' is still a generating set for X but with size $|X'| = |X| - 1$.

Let S be the set of unit intervals that is obtained by a sequence of reduction steps from X such that no more reduction step can be applied to S. Then by Lemma 3.3.17 S has no dependent subset and it must be an isolated sequence. Through a series of technical lemmas, Franzblau at al. further show that if the reduction at every step is done on a minimally dependent unit interval then the isolated sequence S, that is obtained at the end of algorithm and is not necessarily a subset of X, can be translated back into an isolated sequence that is a subset of X and is of the same size

[37, Theorem 4.6]. Therefore for any set of unit intervals X, their algorithm obtains an isolated sequence of X of size $|S|$ and a generating set of X of size $|S|$, proving Győri's theorem.

Theorem 3.3.18 (Győri's Theorem [50])**.** *Interval matrices are firm.*

Chapter 4

Minimally imperfect subgraphs of rectangle cover graphs and the stretching operation

In this chapter, we investigate how odd holes and odd antiholes can appear in the rectangle cover graph of binary matrices. Our aim is to make a small step towards the characterisation of necessary submatrices that lead to the appearance of imperfect subgraphs in rectangle cover graphs.

First, we show that rectangle cover graphs can only contain odd antiholes if they also contain odd holes and hence forbidding odd antiholes for the perfection of rectangle cover graphs is unnecessary. Then, we fully characterise the submatrices that lead to the appearance of 5-holes and show that P_5-free rectangle cover graphs are perfect.

Afterwards, we define simplicial 1s and an operation to remove them. We also introduce an operation called stretching and analyse several versions of it with respect to the preservation of firmness and superfirmness.

In the final section of this chapter, we look at minimally non-superfirm matrices. A binary matrix is called *minimally non-superfirm (mnsf)* if it is not superfirm but all of its proper submatrices are. We prove several matrix families to be mnsf and we analyse their rectangle cover graphs. We then show that every mnsf matrix that is totally balanced is firm and we conjecture that every mnsf matrix is firm.

4.1 Odd antiholes in rectangle cover graphs

Let $\mathbf{X}$ be a binary matrix and let $\mathcal{G}(\mathbf{X})$ be its rectangle cover graph. For a subgraph H of $\mathcal{G}(\mathbf{X})$ induced by some vertex set $V(H) \subseteq \operatorname{supp}_1(\mathbf{X})$, let us look at the submatrix of $\mathbf{X}$ indexed by $I \times J$, where $I = \{i : (i,j) \in V(H)\}$ and $J = \{j : (i,j) \in V(H)\}$. A key idea from Lubiw in [77] is to observe that by duplicating rows and columns of the submatrix indexed by $I \times J$, we can obtain a $|V(H)| \times |V(H)|$ matrix $\mathbf{X}'$ whose rectangle cover graph has a subgraph H' such that no two vertices of H' are in the same row or column, and H' is isomorphic to H. Recall that rectangle cover graphs according to our drawing convention can have three types of edges: horizontal, vertical and diagonal. By duplicating rows and columns of submatrix $I \times J$, we can 'transform' every horizontal and vertical edge of H into a diagonal edge as shown in Figure 4.1. Therefore, all the edges between vertices of H' are diagonal edges in $\mathcal{G}(\mathbf{X}')$.

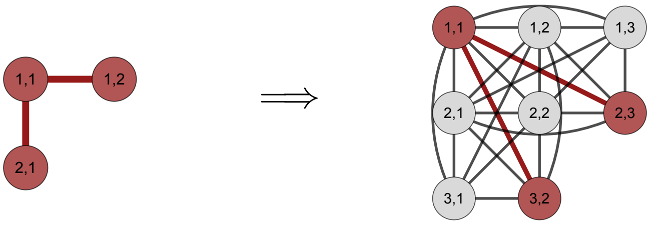


Figure 4.1: A path on three vertices with a horizontal and a vertical edge is transformed into a 3×3 submatrix by row-column duplication which has the path with two diagonal edges

Let us first present a theorem of Lubiw from [77] which inspires the proof of many of our results. Recall that $\mathbf{C}_3$ is the 3×3 cycle matrix.

Theorem 4.1.1 ([77, Theorem 6.3]). *If a binary matrix has no $\mathbf{C}_3$ submatrix then its rectangle cover graph has no odd antihole of size 7 or more.*

Proof. Suppose that $\mathbf{X}$ has no $\mathbf{C}_3$ submatrix but its rectangle cover graph $\mathcal{G}(\mathbf{X})$ has an antihole A with $|V(A)| = n \geq 7$ and odd. Consider the submatrix of $\mathbf{X}$ indexed by $\{i : (i,j) \in V(A)\} \times \{j : (i,j) \in V(A)\}$. By duplicating rows and columns of this submatrix, we may assume to have an $n \times n$ matrix $\mathbf{Y}$ whose rectangle cover graph contains A and that no two of the vertices of A are in the same row or column. Note that row and column duplication cannot introduce $\mathbf{C}_3$ submatrices. Permute $\mathbf{Y}$ so the vertices of A appear in order on the main diagonal. Then $\mathbf{Y}$ must be of the form,

$$\mathbf{Y} = \begin{bmatrix} 1 & \ast & 1 & 1 & \dots & 1 & \ast \\ \ast & 1 & \ast & 1 & & 1 & 1 \\ 1 & \ast & 1 & \ast & & 1 & 1 \\ 1 & 1 & \ast & 1 & & 1 & 1 \\ \vdots & & & & \ddots & & \vdots \\ 1 & 1 & 1 & 1 & & 1 & \ast \\ \ast & 1 & 1 & 1 & \dots & \ast & 1 \end{bmatrix}. \tag{4.1.1}$$

Each 1 on the diagonal is in a common rectangle with all the other 1s on the diagonal that are not directly below or above it ($y_{n,n}$ is considered directly above $y_{1,1}$). To ensure that 1s on the diagonal that are directly below or above each other are not contained in a common rectangle, at least one of $y_{i,i+1}$, $y_{i+1,i}$ for $i \in [n]$ needs to be a 0, where addition of the indices is modulo n.

Without loss of generality assume that $y_{1,2} = 0$. Now suppose that $y_{2,3} = 0$. If $y_{6,5} = 0$, then the submatrix formed by rows $1, 2, 6$ and columns $2, 3, 5$ is $\mathbf{C}_3$. If $y_{5,6} = 0$, then the submatrix formed by rows $1, 2, 5$ and columns $2, 3, 6$ is $\mathbf{C}_3$. Hence $y_{2,3} \neq 0$ and it must be $y_{3,2} = 0$. In general, this shows that $y_{i,i+1}$ and $y_{i+1,i+2}$ cannot be both 0, and the zeros must zig-zag as shown below,

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 & 1 & 1 & \dots & 1 & \ast \\ 1 & 1 & 1 & 1 & & 1 & 1 \\ 1 & 0 & 1 & 0 & & 1 & 1 \\ 1 & 1 & 1 & 1 & & 1 & 1 \\ \vdots & & & & \ddots & & \vdots \\ 1 & 1 & 1 & 1 & & 1 & \ast \\ \ast & 1 & 1 & 1 & \dots & \ast & 1 \end{bmatrix}.$$

But then since n is odd, this is impossible and for some i we have $y_{i,i+1} = y_{i+1,i+2} = 0$ and so $\mathbf{C}_3$ appears. □

This shows that totally balanced matrices cannot have odd antiholes of size 7 or more in their rectangle cover graph. By a simple observation we can slightly strengthen Lubiw's Theorem.

Lemma 4.1.2. *Let $\mathbf{X}$ be a binary matrix. If $\mathcal{G}(\mathbf{X})$ contains an odd antihole of size 7 or more, then $\mathbf{X}$ has at least one of the following 4×4 submatrices,*

$$\begin{bmatrix} \mathbf{C}_3 & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{C}_3 & \mathbf{1} \\ \mathbf{1}^\top & 1 \end{bmatrix}. \tag{4.1.2}$$

Proof. The proof builds on the proof of Theorem 4.1.1, so let $\mathbf{Y}$ be as in Equation (4.1.1) and without loss of generality assume that the two consecutive zeros of the submatrix $\mathbf{C}_3$ that $\mathbf{Y}$ has by Theorem 4.1.1 are $y_{1,2} = y_{2,3} = 0$,

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & \\ \ast & 1 & 0 & 1 & 1 & 1 & \\ 1 & \ast & 1 & \ast & 1 & 1 & \\ 1 & 1 & \ast & 1 & \ast & 1 & \\ 1 & 1 & 1 & \ast & 1 & \ast & \\ 1 & 1 & 1 & 1 & \ast & 1 & \\ & & & & & & \ddots \end{bmatrix}.$$

We must have $y_{6,5} + y_{5,6} \leq 1$. If $y_{6,5} + y_{5,6} = 0$ then the submatrix formed by $\{1,2,5,6\} \times \{2,3,5,6\}$ is the left matrix in Equation (4.1.2). While if $y_{6,5} + y_{5,6} = 1$ the same submatrix is the right matrix in Equation (4.1.2). □

The first matrix in Equation (4.1.2) is just $\bar{\mathbf{I}}_4$, which is shown to be a non-firm matrix with $\mathfrak{i}(\bar{\mathbf{I}}_4) = 3 < \mathfrak{br}(\bar{\mathbf{I}}_4) = 4$ in Example 1.2.1. While the second matrix in Equation (4.1.2) is firm, both matrices contain the proper submatrices $\mathbf{H}_3 = [\mathbf{1}, \mathbf{C}_3]$ and $\mathbf{H}_3^\top$. $\mathbf{H}_3$ is a non-superfirm matrix, as it has three 5-holes in its rectangle cover graph as shown in Figure 3.4. Therefore, a matrix can only have an odd antihole of size 7 or larger in its rectangle cover graph if it also has a proper submatrix whose rectangle cover graph contains 5-holes. Thus together with the fact that a 5-antihole is just a 5-hole, Lemma 4.1.2 implies the following interesting result.

Theorem 4.1.3. *A binary matrix is superfirm if and only if it has no odd holes in its rectangle cover graph.*

This theorem shows that to characterise superfirm matrices in terms of forbidden submatrices, we need only focus on the characterisation of matrices that have odd holes in the rectangle cover graph and every minimally non-superfirm matrix is not superfirm because of an odd hole.

Example 4.1.4. *The smallest matrix that we have found so far with a* 7*-antihole in its rectangle cover graph is shown in Figure 4.2. Observe that it contains* $\left[\begin{smallmatrix} \mathbf{C}_3 & \mathbf{1} \\ \mathbf{1}^\top & 1 \end{smallmatrix}\right]$ *as a proper submatrix, hence its rectangle cover graph has at least six* 5*-holes. In addition, it contains* $\mathbf{M}_3$ *as a proper submatrix as well, so it also has a* 7*-hole.*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & & \\ 1 & 1 & & 1 & \\ 1 & & 1 & 1 & 1 \\ 1 & & & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{1}^\top & 1 \\ \mathbf{1} & \mathbf{C}_3 & \boldsymbol{e}_3 \\ 1 & \boldsymbol{e}_3^\top & 1 \end{bmatrix}$$

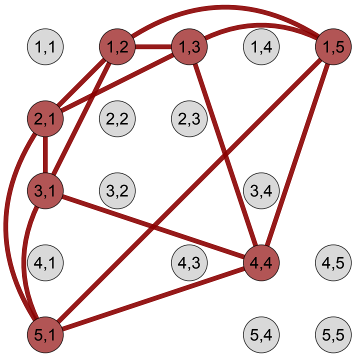


Figure 4.2: A matrix with a 7-antihole in its rectangle cover graph

4.2 Holes and paths in rectangle cover graphs

In this section, we investigate how holes can appear in rectangle cover graphs. Theorem 3.3.9 tells us that a rectangle cover graph is chordal if and only if it has no $\mathbf{D}_3$ and no $\mathbf{C}_n$ submatrices for any $n \geq 3$. Therefore, for a binary matrix to have a hole in its rectangle cover graph it must have a $\mathbf{D}_3$ or $\mathbf{C}_n$ submatrix. In addition, Theorem 3.3.10 tells us that a $\mathbf{D}_3$-free binary matrix is superfirm if and only if it has no $\mathbf{M}_n$ submatrix for any $n \geq 4$. Therefore, if a $\mathbf{D}_3$-free matrix has some odd holes in its rectangle cover graph it must have an $\mathbf{M}_n$ submatrix. We summarise some observations about holes in rectangle cover graphs and matrices that contain them.

Observation 4.2.1. *Let C be a hole in a rectangle cover graph of a binary matrix and let $\mathbf{X}$ be the submatrix indexed by $\{i : (i, j) \in C\} \times \{j : (i, j) \in C\}$.*

1. *$\mathbf{X}$ has a $\mathbf{D}_3$ or $\mathbf{C}_n$ submatrix for some $n \geq 3$ by Theorem 3.3.8.*

2. *If C is an odd-hole then $\mathbf{X}$ has a $\mathbf{D}_3$ or $\mathbf{M}_n$ submatrix for some $n \geq 4$ by Theorem 3.3.10.*

3. *If C is an odd-hole, $\mathbf{X}$ has at least one 2×2 rectangle otherwise it is linear (much weaker condition than 2. but worth remembering).*

4. *From each rectangle of $\mathbf{X}$ (which includes rows and columns), C contains at most two vertices otherwise a chord appears.*

5. *$\mathbf{X}$ has at least two 1s in each row and column, as otherwise if column j only has a single 1 at (i, j), $(i, j) \in C$ and the two neighbours of (i, j) in C are also from row i, which contradicts observation 4.*

6. *If $|C| = 2k$ or $|C| = 2k - 1$, then $\mathbf{X}$ is of dimension at least $k \times k$ by 4.*

7. *If $|C| = 2k$ then $|\operatorname{supp}_1(\mathbf{X})| \geq 2k$, while if $|C| = 2k-1$ then $|\operatorname{supp}_1(\mathbf{X})| \geq 2k+1$ by 3. and 4.*

8. *C cannot have two consecutive vertical edges as otherwise C contains three vertices from a column contradicting 4. The same holds for horizontal edges.*

9. *If C only has vertical and horizontal edges, then these must alternate by 8. and hence C is an even cycle. Thus, if C is an odd hole it has a diagonal edge.*

10. *If C has n_v vertical edges, n_h horizontal edges and n_d diagonal edges, then $\mathbf{X}$ is of dimension $(n_v + n_d) \times (n_h + n_d)$.*

These observations imply that matrices with at most six 1s, or only two rows, or only two columns cannot have an odd hole in their rectangle cover graph. Therefore, any matrix with at most six 1s or of dimension $2 \times n$ or $m \times 2$ is superfirm.

The observations also show that the dimension and number of 1s of $\mathbf{M}_n$ are the minimum possible for a matrix with a $2n - 1$-hole in the rectangle cover graph.

In the next sections, we use these observations to investigate which matrices cause the appearance of 4- and 5-holes, and paths on 4 and 5 vertices.

4.2.1 4-holes

We have seen in Section 3.3.3 that forbidding $\mathbf{D}_3$ matrices leads to a firm class of matrices. Furthermore, in Figure 3.6 we show that the rectangle cover graph of $\mathbf{D}_3$ contains a 4-hole. It turns out that $\mathbf{D}_3$ is the only submatrix responsible for the appearance of 4-holes in rectangle cover graphs as the next lemma shows.

Lemma 4.2.2. *The rectangle cover graph of a binary matrix $\mathbf{X}$ contains a 4-hole if and only if $\mathbf{X}$ has a $\mathbf{D}_3$ submatrix.*

Proof. $\mathcal{G}(\mathbf{D}_3)$ clearly contains a 4-hole as shown in Figure 3.6.

For the other direction, suppose that $\mathbf{X}$ has no submatrix $\mathbf{D}_3$ but $\mathcal{G}(\mathbf{X})$ has a 4-hole. Consider the submatrix indexed by the rows and columns of the 4-hole and duplicate and permute its rows and columns if needed to get a 4×4 matrix $\mathbf{Y}$ whose rectangle cover graph contains a 4-hole and the vertices of it appear on the main diagonal consecutively. Note that row and column duplication cannot introduce $\mathbf{D}_3$ submatrices. Then $\mathbf{Y}$ must be of the form,

$$\mathbf{Y} = \begin{bmatrix} 1 & 1 & * & 1 \\ 1 & 1 & 1 & * \\ * & 1 & 1 & 1 \\ 1 & * & 1 & 1 \end{bmatrix}.$$

Inequalities $y_{1,3} + y_{3,1} \leq 1$ and $y_{2,4} + y_{4,2} \leq 1$ must hold otherwise a chord appears between the vertices of the 4-hole. While inequalities $y_{1,3} + y_{3,1} \geq 1$ and $y_{2,4} + y_{4,2} \geq 1$ must hold otherwise a $\mathbf{D}_3$ submatrix appears.

Without loss of generality let $y_{1,3} = 1$. Then if $y_{2,4} = 1$, submatrix $\{2,3,4\} \times \{1,2,3\}$ is $\mathbf{D}_3$. While, if $y_{2,4} = 0$, submatrix $\{1,3,4\} \times \{1,2,4\}$ is $\mathbf{D}_3$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Therefore, $\mathbf{D}_3$ appears in any case, a contradiction. □

4.2.2 5-holes

From Theorem 3.3.10 we know that $\mathbf{D}_3$-free matrices can have 7-holes and larger in their rectangle cover graph through matrices $\mathbf{M}_n$. Can $\mathbf{D}_3$-free matrices also have 5-holes? An almost identical proof to that of Lemma 4.2.2 with a few more cases to write out shows that 5-holes cannot appear without a $\mathbf{D}_3$ submatrix. We however omit the proof of this, because it will be implied by the theorem that we state and prove soon further below.

Let $\mathbf{K}_5$ be the 5×5 matrix below. The rectangle cover graph of $\mathbf{K}_5$ then contains a 5-hole as shown in Figure 4.3.

$$\mathbf{K}_5 = \begin{bmatrix} 1 & 1 & & & 1 \\ 1 & 1 & 1 & & \\ & 1 & 1 & 1 & \\ & & 1 & 1 & 1 \\ 1 & & & 1 & 1 \end{bmatrix}.$$

The following theorem tells us which are the necessary and sufficient submatrices for a binary matrix to have a 5-hole in its rectangle cover graph.

Theorem 4.2.3. *The rectangle cover graph of a binary matrix contains a* 5*-hole if and only if the matrix has at least one of* $\mathbf{D}_4$, $\mathbf{H}_3$, $\mathbf{H}_3^\top$, $\mathbf{K}_5$ *as a submatrix.*

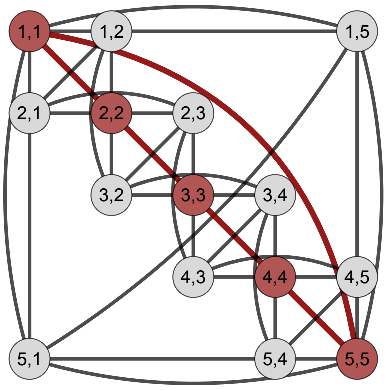


Figure 4.3: $\mathcal{G}(\mathbf{K}_5)$ and its 5-hole highlighted

Proof. Recall that as shown in Figures 3.2, 3.4, 4.3, all of $\mathbf{D}_4$, $\mathbf{H}_3$ and $\mathbf{K}_5$ have a 5-hole in their rectangle cover graph.

For the reverse, suppose that $\mathbf{X}$ has a 5-hole in its rectangle cover graph $\mathcal{G}(\mathbf{X})$ but it has none of the submatrices $\mathbf{D}_4$, $\mathbf{H}_3$, $\mathbf{H}_3^\top$ and $\mathbf{K}_5$. Consider the submatrix indexed by the rows and columns of the 5-hole and duplicate and permute its rows and columns if needed to get a 5 × 5 matrix $\mathbf{Y}$ whose rectangle cover graph contains a 5-hole whose vertices appear on the main diagonal consecutively. Note that none of the submatrices concerned can be created through row column duplication. Then $\mathbf{Y}$ is of the form,

$$\mathbf{Y} = \begin{bmatrix} 1 & 1 & * & * & 1 \\ 1 & 1 & 1 & * & * \\ * & 1 & 1 & 1 & * \\ * & * & 1 & 1 & 1 \\ 1 & * & * & 1 & 1 \end{bmatrix},$$

where $y_{i,j} + y_{j,i} \leq 1$ for $(i,j) \in \{(1,3),(1,4),(2,4),(2,5),(3,5)\}$ to ensure that the 5-hole is chordless.

If all $*$'s are set to 0, then $\mathbf{Y} = \mathbf{K}_5$, hence at least one of them is a 1. Without loss of generality, $y_{1,3} = 1$. If $y_{1,4} = y_{3,5} = y_{5,3} = 0$, then submatrix $\{1,3,4,5\} \times \{3,4,5\}$ is $\mathbf{H}_3^\top$ as highlighted below on the left. Hence at least one of them is equal to 1. We go into case enumeration from here.

$$\begin{bmatrix} 1 & 1 & 1 & * & 1 \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & * \\ * & * & 1 & 1 & 1 \\ 1 & * & * & 1 & 1 \end{bmatrix} \rightarrow (a) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & * \\ 0 & 1 & 1 & 1 & * \\ 0 & * & 1 & 1 & 1 \\ 1 & * & * & 1 & 1 \end{bmatrix}, \quad (b) \begin{bmatrix} 1 & 1 & 1 & * & 1 \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & 1 \\ * & * & 1 & 1 & 1 \\ 1 & * & 0 & 1 & 1 \end{bmatrix}$$

(a) Let $y_{1,4} = 1$. If $y_{2,5} = y_{5,3} = 0$, then $\mathbf{H}_3^\top$ appears, hence at least one of $y_{2,5}, y_{5,3}$ is equal to 1.

(b) Let $y_{3,5} = 1$. If $y_{1,4} = y_{4,1} = 0$, then $\mathbf{H}_3$ appears, hence exactly one of them is equal to 1.

(c) If $y_{5,3} = 1$ then $\mathbf{Y}$ with a permutation of rows $\{3, 4, 5, 1, 2\}$ and columns $\{3, 4, 5, 1, 2\}$ is just the transpose of case (a), hence needs no separate treatment.

In case (b), we have (b1) $y_{4,1} = 1$ or (b2) $y_{1,4} = 1$. In case (b1), $\mathbf{H}_3$ is present.

$$(b) \rightarrow \quad (b1) \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & 1 \\ 1 & * & 1 & 1 & 1 \\ 1 & * & 0 & 1 & 1 \end{bmatrix}, \ (b2) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & 1 \\ 0 & * & 1 & 1 & 1 \\ 1 & * & 0 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix},$$

In case (b2), submatrix $\{1, 2, 3, 5\} \times \{1, 3, 4\}$ or $\{1, 2, 3, 5\} \times \{1, 3, 5\}$ is $\mathbf{H}_3^\top$ unless $y_{2,4} = y_{2,5} = 1$. But then submatrix $\{2, 3, 4, 5\} \times \{1, 2, 3, 4\}$ is $\mathbf{D}_4$.

In case (a), we have (a1) $y_{5,3} = 1$ or (a2) $y_{2,5} = 1$.

$$(a) \rightarrow (a1) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & 0 \\ 0 & * & 1 & 1 & 1 \\ 1 & * & 1 & 1 & 1 \end{bmatrix} \rightarrow (a1.i) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & * \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & * & 1 & 1 & 1 \end{bmatrix}, (a1.ii) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & * & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & * & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

In (a1), if $y_{2,4} = y_{5,2} = 0$ then $\mathbf{H}_3^\top$ appears, so we must have either (a1.i) $y_{2,4} = 1$ or (a1.ii) $y_{5,2} = 1$. In (a1.i), since $y_{2,5} + y_{5,2} \leq 1$, submatrix $\{2, 3, 4, 5\} \times \{1, 2, 3, 5\}$ is $\mathbf{D}_4$. Similarly in (a.1.ii), since $y_{2,4} + y_{4,2} \leq 1$, submatrix $\{2, 3, 4, 5\} \times \{1, 2, 4, 5\}$ is $\mathbf{D}_4$.

In case (a2), if $y_{4,2} = 1$ then $\mathbf{H}_3^\top$ appears, so we must have $y_{4,2} = 0$. But then since $y_{3,5} + y_{5,3} \leq 1$, $\mathbf{D}_4$ appears in the submatrix formed by rows $\{1, 3, 4, 5\}$ and columns $\{1, 2, 3, 4\}$ or $\{1, 2, 4, 5\}$.

$$(a2) \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & * & 1 \\ 0 & 1 & 1 & 1 & * \\ 0 & * & 1 & 1 & 1 \\ 1 & 0 & * & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & * & 1 \\ 0 & 1 & 1 & 1 & * \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & * & 1 & 1 \end{bmatrix},$$

Therefore, in all cases at least one of $\mathbf{H}_3, \mathbf{H}_3^\top, \mathbf{D}_4$ or $\mathbf{K}_5$ appears as a submatrix. □

Observe that all necessary submatrices of this theorem contain a $\mathbf{D}_3$ submatrix, hence 5-holes can only appear along with 4-holes in rectangle cover graphs. By this we learn that the firm class of $\mathbf{D}_3$-free matrices cannot have 5-holes in their rectangle cover graphs.

By our observations at the beginning of the section, the above theorem also shows that the smallest non-superfirm matrix is $\mathbf{H}_3$ of dimension 3×4 and its transpose. In addition, note that while $\mathbf{K}_5$ is necessary to be included in the above theorem, it contains several $\mathbf{M}_4$ as proper submatrices (for instance submatrix $\{1,2,3,4\} \times \{1,3,4,5\}$ is $\mathbf{M}_4$). Since $\mathbf{M}_4$ is not superfirm, $\mathbf{K}_5$ cannot be minimally non-superfirm.

4.2.3 Induced paths

Let P_n denote the path graph on n vertices. Graphs which do not contain P_4 as an induced subgraph are called *cographs.* Observe that the complement of P_4 is P_4 itself. Since any odd-hole and odd-antihole contains P_4 as an induced subgraph, cographs form an important subset of perfect graphs. The following theorem shows which matrices have cograph rectangle cover graphs.

Theorem 4.2.4. *The rectangle cover graph of a binary matrix* $\mathbf{X}$ *contains* P_4 *as an induced subgraph if and only if* $\mathbf{X}$ *has at least one of the following submatrices,*

$$\begin{bmatrix} v_1 & v_2 & 0 \\ 0 & v_3 & v_4 \end{bmatrix}, \qquad \begin{bmatrix} v_1 & 0 \\ v_2 & v_3 \\ 0 & v_4 \end{bmatrix}, \qquad \begin{bmatrix} v_1 & v_2 & 1 \\ 0 & 1 & v_3 \\ 0 & 0 & v_4 \end{bmatrix}, \tag{4.2.1}$$

where $v_i = 1$ *for all* $i \in [4]$ *and indicate the vertices of the induced* P_4 *subgraph in the matrices' rectangle cover graph.*

Proof. Let us refer to the three matrices considered by $\mathbf{X}_1, \mathbf{X}_1^\top, \mathbf{X}_2$ from left to right. Clearly all three matrices have an induced P_4 in their rectangle cover graph.

For the reverse, we proceed by the exact same proof method as for the previous theorems in this section. So suppose the rectangle cover graph of a matrix has an induced P_4 but no $\mathbf{X}_1, \mathbf{X}_1^\top, \mathbf{X}_2$ submatrices and consider its submatrix that contains P_4 and by permutations and row-column duplications assume that this submatrix $\mathbf{Y}$ is 4×4 and the vertices of P_4 are on the main diagonal consecutively,

$$\begin{bmatrix} 1 & 1 & * & * \\ 1 & 1 & 1 & * \\ * & 1 & 1 & 1 \\ * & * & 1 & 1 \end{bmatrix}.$$

As before, the considered submatrices cannot be introduced by row column duplication. So that P_4 has no chord and $\mathbf{X}_1$ and $\mathbf{X}_1^\top$ do not appear, we must have $y_{1,3} + y_{3,1} = 1$, $y_{1,4} + y_{4,1} = 1$ and $y_{2,4} + y_{4,2} = 1$. Without loss of generality, let $y_{1,3} = 0$ and $y_{3,1} = 1$. Then we must also have $y_{4,1} = y_{4,2} = 1$ so that $\mathbf{X}_1$ does not appear. But then submatrix $\{2,3,4\} \times \{2,3,4\}$ is $\mathbf{X}_2$. □

Therefore, binary matrices that do not have any of the above three submatrices are superfirm. Is this a new class of superfirm matrices? In general, cographs are not chordal because, for instance a 4-hole does not have an induced P_4. However, among rectangle cover graphs, by forbidding the three matrices of Theorem 4.2.4 we also forbid $\mathbf{D}_3$ and $\mathbf{C}_n$ for all $n \geq 3$. Since $\mathbf{D}_3$-$\mathbf{C}_n$-free matrices have chordal rectangle cover graphs by Theorem 3.3.9, cograph rectangle cover graphs are chordal.

What can we say about induced P_5's in rectangle cover graphs? The enumeration gets messier, so the following is the last of our 'enumeration-type' results.

Theorem 4.2.5. *The rectangle cover graph of a binary matrix* $\mathbf{X}$ *has an induced* P_5 *subgraph if and only if* $\mathbf{X}$ *has one of the submatrices,*

$$\begin{bmatrix} v_1 & v_2 & 0 \\ 0 & v_3 & v_4 \\ * & 0 & v_5 \end{bmatrix}, \begin{bmatrix} v_1 & v_2 & 1 & * \\ 0 & 1 & v_3 & 0 \\ 0 & 0 & v_4 & v_5 \end{bmatrix}, \begin{bmatrix} v_1 & 0 & 0 \\ v_2 & 1 & 0 \\ 1 & v_3 & v_4 \\ * & 0 & v_5 \end{bmatrix}, \begin{bmatrix} v_1 & v_2 & 1 & 1 \\ 0 & 1 & v_3 & 1 \\ 0 & 0 & 1 & v_4 \\ 0 & 0 & 0 & v_5 \end{bmatrix}, \tag{4.2.2}$$

where entries marked with $*$ *can be either* 0 *or* 1 *and* $v_i = 1$ *for all* i *denoting the vertices of* P_5. *(Note that the leftmost and rightmost matrices can be permuted to be symmetric, hence with* $* \in \{0,1\}$ *we specify* 7 *matrices in total.)*

Proof. Let us refer to the matrices considered by $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_2^\top, \mathbf{X}_3$ from left to right. Clearly all have an induced P_5 in their rectangle cover graph as indicated.

For the reverse, we proceed by the exact same proof method as for the previous theorems in this section, so we suppose that 5×5 $\mathbf{Y}$ has an induced P_5 in its rectangle cover graph with the vertices of P_5 on the main diagonal, and $\mathbf{Y}$ has none of the considered submatrices,

$$\begin{bmatrix} 1 & 1 & * & * & * \\ 1 & 1 & 1 & * & * \\ * & 1 & 1 & 1 & * \\ * & * & 1 & 1 & 1 \\ * & * & * & 1 & 1 \end{bmatrix}.$$

So that the diagonal 1s form P_5, we must have at most one of the two $*$'s symmetric about the diagonal equal to 1. Observe that if $y_{1,3} + y_{3,1} = 0$, then if any of $y_{2,4}, y_{4,2}$

is 0 then $\mathbf{X}_1$ appears. Hence $y_{1,3} + y_{3,1} = 1$. Similarly, if $y_{3,5} + y_{5,4} = 0$, then if any of $y_{2,4}, y_{4,2}$ is 0 then $\mathbf{X}_1$ appears. Hence $y_{3,5} + y_{5,3} = 1$. So we have two cases: (a) $y_{1,3} = y_{3,5} = 1$ and (b) $y_{1,3} = y_{5,3} = 1$ as the other two cases are symmetric.

$$(a) \begin{bmatrix} 1 & 1 & 1 & * & * \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & 1 \\ * & * & 1 & 1 & 1 \\ * & * & 0 & 1 & 1 \end{bmatrix} \qquad (b) \begin{bmatrix} 1 & 1 & 1 & * & * \\ 1 & 1 & 1 & * & * \\ 0 & 1 & 1 & 1 & 0 \\ * & * & 1 & 1 & 1 \\ * & * & 1 & 1 & 1 \end{bmatrix}$$

In case (a), if $y_{2,4} = 0$, then $\{2,3,5\} \times \{1,3,4\}$ is $\mathbf{X}_1$, so $y_{4,2} = 1$. But then again if $y_{2,5} = 0$ then $\{2,3,4\} \times \{1,2,5\}$ is $\mathbf{X}_1$, or if $y_{1,4} = 0$ then $\{1,2,4\} \times \{2,3,4\}$ is $\mathbf{X}_1$, so we must have $y_{1,4} = y_{2,5} = 1$. Now if $y_{5,1} = 0$, then $\{1,2,3,4\} \times \{1,2,3,4\}$ is $\mathbf{X}_3$ so $y_{5,1} = 1$. But then $\{1,3,5\} \times \{1,3,5\}$ is $\mathbf{X}_1$.

In case (b), if $y_{2,3} + y_{3,2} = 0$ then $\{2,3,4\} \times \{2,4,5\}$ is $\mathbf{X}_1$, so we must have $y_{2,3} + y_{3,2} = 1$ and without loss of generality let $y_{2,3} = 1$. But then if $y_{1,4} = 0$, $\{1,3,4\} \times \{1,2,4\}$ is $\mathbf{X}_1$, so $y_{1,4} = 1$. But then $\{2,3,4\} \times \{1,2,3,5\}$ is $\mathbf{X}_2$.

Therefore, in any case one of the matrices appears and they are necessary for the appearance of an induced P_5 in rectangle cover graphs. □

Note that P_5-free graphs cannot contain holes of size greater than 5. The complement of P_5 is the house graph, which is a 5-cycle with exactly one chord. This helps to see that odd antiholes and the 5-hole are P_5-free, hence in general P_5-free graphs are not perfect. However, by forbidding the leftmost matrix in Theorem 4.2.5 we also forbid matrix $\mathbf{C}_3$ and then by Theorem 4.1.1 P_5-free rectangle cover graphs cannot contain an odd antihole of size 7 or more. In addition, $\mathbf{H}_3$, $\mathbf{H}_3^\top$ and $\mathbf{K}_5$ all have a $\mathbf{C}_3$ submatrix hence these are also forbidden by the leftmost matrix in Theorem 4.2.5. Furthermore, $\mathbf{D}_4$ has the centre matrices in Theorem 4.2.5 as submatrices, hence $\mathbf{D}_4$ is also forbidden when forbidding P_5. Therefore, by the submatrix characterisation of 5-holes in Theorem 4.2.3, forbidding P_5 forbids all submatrices that are necessary for the appearance of 5-holes, odd-antiholes and odd-holes of size 7 or larger. This together with the Strong Perfect Graph Theorem leads to the following result.

Theorem 4.2.6. *If a rectangle cover graph is P_5-free then it is perfect.*

4.3 Simplicial 1s and stretching

In this section, we introduce simplicial 1s and a removal operation for them that decreases both the Boolean rank and the isolation number by exactly 1. Then we introduce an operation related to simplicial 1s, which we call 'stretching'. The stretching operation will be a crucial part of almost all the results that we prove in the rest of Part I of this thesis.

Simplicial 1s. Let $\mathbf{X}$ be a generalised binary matrix. We say $(\ell, k) \in \mathrm{supp}_1(\mathbf{X})$ is a *simplicial* 1 of $\mathbf{X}$ if $I \times J$ with $I = \{i : x_{i,k} \in \{1, ?\}\}$ and $J = \{j : x_{\ell,j} \in \{1, ?\}\}$ satisfies $I \times J \subseteq \mathrm{supp}_1(\mathbf{X}) \cup \mathrm{supp}_?(\mathbf{X})$, that is $I \times J$ is a rectangle of $\mathbf{X}$. Note that $I \times J$ is a maximal rectangle and the only maximal rectangle of $\mathbf{X}$ that covers the simplicial 1 at (ℓ, k), because if any other rectangle $I' \times J'$ covers (ℓ, k) then it has $I' \subseteq I$ and $J' \subseteq J$, and so it can only be maximal if it is equal to $I \times J$. To *remove the simplicial* 1 at (ℓ, k) of $\mathbf{X}$ we delete row ℓ and column k and set all remaining entries that are in $I \times J$ to ?s.

Lemma 4.3.1. *If $\boldsymbol{X}'$ is obtained by removing a simplicial* 1 *of a generalised binary matrix $\boldsymbol{X}$, then $\mathfrak{i}(\boldsymbol{X}) = \mathfrak{i}(\boldsymbol{X}') + 1$ and $\mathfrak{br}(\boldsymbol{X}) = \mathfrak{br}(\boldsymbol{X}') + 1$.*

Proof. Let (ℓ, k) be the simplicial 1 and $I \times J$ its unique maximal rectangle. If S' is a maximum isolated set and $\mathcal{R}'$ is a minimum rectangle cover of $\mathbf{X}'$, then $S' \cup \{(\ell, k)\}$ is a feasible isolated set and $\mathcal{R}' \cup (I \times J)$ is a feasible rectangle cover of $\mathbf{X}$. Conversely, if S is a maximum isolated set of $\mathbf{X}$, then we must have $S \cap (I \times J) = \{(i^*, j^*)\}$ for some $(i^*, j^*) \in I \times J$, as otherwise $S \cup \{(\ell, k)\}$ would be a larger isolated set of $\mathbf{X}$. So $S \setminus \{(i^*, j^*)\}$ is a feasible isolated set of $\mathbf{X}'$. As (ℓ, k) is a simplicial 1, and $I \times J$ is the only maximal rectangle in $\mathbf{X}$ that covers (ℓ, k), we may assume that $I \times J$ is used in a minimum cover $\mathcal{R}$ of $\mathbf{X}$. Then $\mathcal{R} \setminus \{I \times J\}$ is a feasible rectangle cover of $\mathbf{X}'$. □

The above lemma naturally holds when removing a simplicial 1 of a standard binary matrix. It is in some way a method to transition from standard binary matrices to generalised binary matrices. In case, after removing a simplicial 1 we get a generalised binary matrix which has a row or column of all ?s, it is useful to observe the following.

Observation 4.3.2. *If $\mathbf{X}$ is a generalised binary matrix with a row that does not have any* 1*s and $\mathbf{Y}$ is obtained from $\mathbf{X}$ by deleting this row, then $\mathfrak{i}(\mathbf{X}) = \mathfrak{i}(\mathbf{Y})$ and $\mathfrak{br}(\mathbf{X}) = \mathfrak{br}(\mathbf{Y})$.*

Our definition of simplicial 1s for a standard binary matrix $\mathbf{X}$ is identical to the definition of *bisimplicial edges* of [41] in the bipartite setting $\mathcal{B}(\mathbf{X})$. The key difference is how we remove a simplicial 1 and transition into generalised binary matrices.

A *simplicial vertex* of a graph is one whose neighbours form a clique. Also observe that if (ℓ, k) is a simplicial 1 of a generalised binary matrix then it is also a simplicial vertex of $\mathcal{G}(\mathbf{X})$. On the other hand, the converse does not necessarily hold. For instance (ℓ, k) of matrix $\mathbf{X}$,

$$\mathbf{X} = \begin{array}{c} \\ \ell \\ i \\ \\ \end{array}\begin{array}{c} \begin{array}{ccc} k & j & t \end{array} \\ \begin{bmatrix} 1 & ? & 1 \\ ? & 0 & 1 \\ 0 & 1 & ? \end{bmatrix} \end{array},$$

is a simplicial vertex of $\mathcal{G}(\mathbf{X})$ as it has two neighbours (ℓ, t) and (i, t) and they form a triangle, but (ℓ, k) is not a simplicial 1 of $\mathbf{X}$ because $\{\ell, i\} \times \{k, j, t\} \not\subseteq \mathrm{supp}_1(\mathbf{X}) \cup \mathrm{supp}_?(\mathbf{X})$.

Stretching. In Section 3.1, we showed that not every induced subgraph of $\mathcal{G}(\mathbf{X})$ corresponds to a submatrix of $\mathbf{X}$, but by turning 1s to ?s we can consider arbitrary induced subgraphs of $\mathcal{G}(\mathbf{X})$ in matrix form. The idea behind the next matrix operation is to expose induced subgraphs of rectangle cover graphs without explicitly setting matrix entries to ?s. This operation will be one of the key ingredients used in Chapter 5 to construct minimally non-firm matrices.

Let $\mathbf{X} \in \{0,1\}^{m \times n}$. By *stretching* $(\ell, k) \in \mathrm{supp}_1(\mathbf{X})$ we get the $(m+1) \times (n+1)$ binary matrix $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ which satisfies

$$\begin{aligned} &\mathcal{S}^{(\ell,k)}(\mathbf{X})_{i,j} = x_{i,j} && i \in [m], j \in [n], \\ &\mathcal{S}^{(\ell,k)}(\mathbf{X})_{i,j} = 1 && (i,j) \in \{(\ell, n+1), (m+1, k), (m+1, n+1)\}, \\ &\mathcal{S}^{(\ell,k)}(\mathbf{X})_{i,j} = 0 && \text{otherwise.} \end{aligned}$$

For instance, if $(m, n) \in \mathrm{supp}_1(\mathbf{X})$ then by stretching (m, n) we obtain

$$\mathcal{S}^{(m,n)}(\mathbf{X}) = \begin{bmatrix} x_{1,1} & \dots & & x_{1,n} & 0 \\ \vdots & \ddots & & \vdots & \vdots \\ & & & x_{m-1,n} & 0 \\ x_{m,1} & \dots & x_{m,n-1} & 1 & 1 \\ 0 & \dots & 0 & 1 & 1 \end{bmatrix}.$$

Stretching (ℓ, k) adds in a simplicial 1 at position $(m+1, n+1)$ whose unique maximal rectangle covers only (ℓ, k) from $\mathrm{supp}_1(\mathbf{X})$. By Lemma 4.3.1, removing the simplicial 1 at $(m+1, n+1)$, we get

$$\begin{aligned} \mathfrak{i}(\mathcal{S}^{(\ell,k)}(\mathbf{X})) &= \mathfrak{i}(\mathbf{X}^{(\ell,k)}) + 1, \\ \mathfrak{br}(\mathcal{S}^{(\ell,k)}(\mathbf{X})) &= \mathfrak{br}(\mathbf{X}^{(\ell,k)}) + 1, \end{aligned}$$

where $\mathbf{X}^{(\ell,k)}$ is a shorter notation for the generalised binary matrix $\mathbf{X}^P$ with $P = \{(\ell, k)\}$.

For a non-empty set $Q \subseteq \operatorname{supp}_1(\mathbf{X})$, the matrix obtained by stretching each 1 in Q is denoted by $\mathcal{S}^Q(\mathbf{X})$. We adopt the convention to stretch 1s in Q in non-decreasing order of row and then column index, so $\mathcal{S}^Q(\mathbf{X})$ may be written in block form as

$$\mathcal{S}^Q(\mathbf{X}) = \begin{bmatrix} \mathbf{X} & \mathbf{U} \\ \mathbf{L} & \mathbf{I}_{|Q|} \end{bmatrix} \tag{4.3.1}$$

where $\mathbf{U}$ is an $m \times |Q|$ matrix with $|Q|$ 1s exactly one in each column that have non-decreasing row index from left to right, $\mathbf{L}$ is an $|Q| \times n$ matrix with $|Q|$ 1s exactly one in each row and $\mathbf{I}_t$ is the $t \times t$ identity matrix. If we wish to then stretch a 1 of matrix $\mathcal{S}^Q(\mathbf{X})$ that is created by stretching Q, we denote that by iterating the stretching operator: $\mathcal{S}^{(\ell,k)}(\mathcal{S}^Q(\mathbf{X}))$.

Let us see how stretching affects firmness and superfirmness.

Lemma 4.3.3. *If* $\mathbf{X}$ *is superfirm, then* $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ *is firm.*

Proof. As $\mathbf{X}$ is superfirm, $\mathcal{G}(\mathbf{X})$ is a perfect graph. The subgraph H of $\mathcal{G}(\mathbf{X})$ that we obtain by deleting vertex (ℓ, k) satisfies $\alpha(H) = \theta(H)$. Since entry $(m+1, n+1)$ of $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ is a simplicial 1, removing it we have $\mathfrak{i}(\mathcal{S}^{(\ell,k)}(\mathbf{X})) = \alpha(H) + 1 = \theta(H) + 1 = \mathfrak{br}(\mathcal{S}^{(\ell,k)}(\mathbf{X}))$.

It is easy to see that any proper submatrix $\mathbf{X}'$ of $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ that is not fully contained in $\mathbf{X}$ has a 1 from row $m+1$ or column $n+1$ and one of those 1s is a simplicial 1. Therefore, after removing that simplicial 1 we get $\mathfrak{i}(\mathbf{X}') = \alpha(H) + 1 = \theta(H) + 1 = \mathfrak{br}(\mathbf{X}')$ for some perfect subgraph H of $\mathcal{G}(\mathbf{X})$. □

We say $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ is obtained by *simplicial stretching* if (ℓ, k) is a simplicial 1 of $\mathbf{X}$.

Lemma 4.3.4. *If* $\mathbf{X}$ *is superfirm and* $(\ell, k) \in \operatorname{supp}_1(\mathbf{X})$ *is a simplicial* 1*, then* $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ *is superfirm. In particular, simplicial stretching preserves superfirmness.*

Proof. Let $\mathbf{X}$ have a simplicial 1 at position (ℓ, k) and let $\mathbf{X}' = \mathcal{S}^{(\ell,k)}(\mathbf{X})$. By assumption $\mathcal{G}(\mathbf{X})$ is perfect and (ℓ, k) is a simplicial vertex of $\mathcal{G}(\mathbf{X})$. Let K denote the unique maximal clique of (ℓ, k) in $\mathcal{G}(\mathbf{X})$. Suppose that $\mathcal{G}(\mathbf{X}')$ is not perfect and let H be a minimally imperfect induced subgraph of $\mathcal{G}(\mathbf{X}')$. Note that H contains at least one vertex from $W = \{(\ell, n+1), (m+1, k), (m+1, n+1)\}$ as otherwise H is a subgraph of $\mathcal{G}(\mathbf{X})$. If $V(H) \subset K \cup W$, then the complement of H, $\overline{H}$ is a bipartite graph with bipartition $V_1 = V(H) \cap K$ and $V_2 = V(H) \cap W$, hence H is perfect by the Weak Perfect Graph Theorem as bipartite graphs are perfect. So H must have at least one vertex from $V(\mathcal{G}(\mathbf{X})) \setminus (K \cup W)$. But then $K \cap V(H)$ is a clique cutset of H which by the Clique Sum Lemma 3.1.10 contradicts the minimally imperfectness of H. □

The above lemma is tight in two ways. First, both simplicial and non-simplicial stretching do not preserve firmness and this property of stretching is the one that will be explored in the next chapter to create minimally non-firm matrices. Second, non-simplicial stretching also does not preserve superfirmness as the next example shows.

Example 4.3.5 (Stretching destroys superfirmness I.)**.** *Observe that matrix* $\mathbf{M}_n$ *(defined in Equation* (3.3.2)*) is obtained by stretching the non-simplicial* 1 *at position* $(n-1, n-1)$ *of the superfirm matrix* $\mathbf{C}_{n-1}$ *whose rectangle cover graph is just a* $2(n-1)$*-hole, so for all* $n \geq 4$*,*

$$\mathbf{M}_n = \mathcal{S}^{(n-1,n-1)}(\mathbf{C}_{n-1}).$$

As we have seen in Figure 3.7 for $n = 4, 5$*,* $\mathbf{M}_n$ *contains a* $2n-1$*-hole. Hence non-simplicial stretching can destroy superfirmness.*

Example 4.3.6 (Stretching destroys superfirmness II.)**.** *Let* $\mathbf{X}$ *be the* 4×4 *interval matrix whose rectangle cover graph is shown on the left hand side of Figure 4.4. Observe that* $\mathcal{G}(\mathbf{X})$ *has a* 6*-hole which is highlighted. Stretching the non-simplicial* 1 *at* $(3, 2)$ *we create a 7-hole as shown on the right hand side of Figure 4.4.*

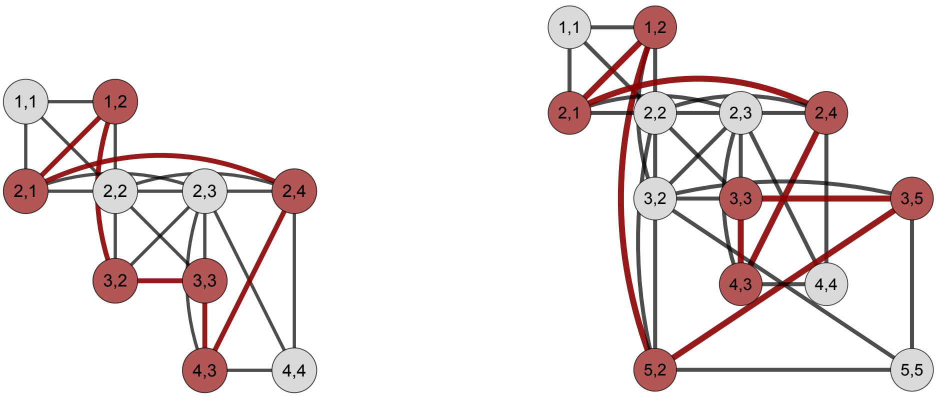


Figure 4.4: Creating a 7-hole by stretching $(3, 2)$

Now let us illustrate a general recipe for how one can destroy superfirmness by non-simplicial stretching. Let $\mathbf{X}$ have an even hole C in $\mathcal{G}(\mathbf{X})$ with $(\ell, j), (\ell, k), (i, k) \in$

$\operatorname{supp}_1(\mathbf{X})$ being three vertices of C. Then we must have $(i,j) \in \operatorname{supp}_0(\mathbf{X})$ and submatrix $\{i,\ell\} \times \{j,k\}$ of $\mathbf{X}$ is the matrix shown below on the left hand side,

$$\begin{array}{c|cc} & j & k \\ \hline i & 0 & 1 \\ \ell & 1 & 1 \end{array} \qquad \overset{\mathcal{S}^{(\ell,k)}}{\Longrightarrow} \qquad \begin{array}{c|ccc} & j & k & n{+}1 \\ \hline i & 0 & 1 & 0 \\ \ell & 1 & 1 & 1 \\ m{+}1 & 0 & 1 & 1 \end{array}. \tag{4.3.2}$$

By stretching the non-simplicial 1 of $\mathbf{X}$ at (ℓ,k) we can replace vertex (ℓ,k) of the even hole C by two new vertices $(m+1,k)$ and $(\ell,n+1)$ to form an odd hole with vertices as shown in Figure 4.5.

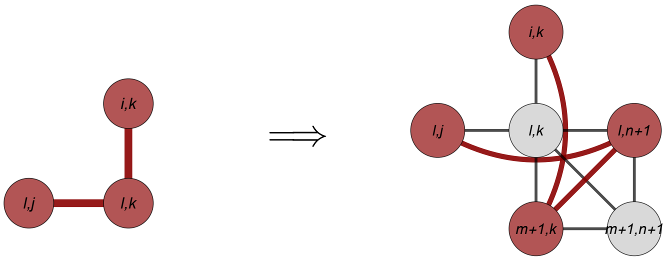


Figure 4.5: Stretching a non-simplicial 1 can destroy superfirmness

Then $(C \setminus \{(\ell,k)\}) \cup \{(m+1,k),(\ell,n+1)\}$ is indeed an odd hole as row ℓ and column k already had two vertices of C, so the new vertices $(m+1,k)$,$(\ell,n+1)$ do not form any chord. Naturally, this kind of stretching could also be applied to a matrix which has an odd-hole and three such vertices of the odd hole to obtain a new even hole.

2-simplicial neighbour stretching. While as argued, non-simplicial stretching does not preserve superfirmness, if we make some more restrictive assumptions on the 1 being stretched, we can show that another kind of stretching preserves superfirmness and firmness.

Definition 4.3.7. *We say* $(i,k) \in \operatorname{supp}_1(\mathbf{X})$ *(or* $(\ell,j) \in \operatorname{supp}_1(\mathbf{X})$*) is a* 2-simplicial neighbour *if*

- *there is only one other non-zero in column* k *(or row* ℓ*) at position* (ℓ,k)*,*
- *and* (ℓ,k) *is a simplicial* 1.

For instance, the shaded 1s of $\mathbf{D}_4$ below are all 2-simplicial neighbours,

$$\mathbf{D}_4 = \begin{bmatrix} 1 & \colorbox{lightgray}{1} & & \\ \colorbox{lightgray}{1} & 1 & 1 & \colorbox{lightgray}{1} \\ & 1 & 1 & 1 \\ & \colorbox{lightgray}{1} & 1 & \end{bmatrix}.$$

Lemma 4.3.8. *If $\mathbf{X}$ is a firm matrix and (i,k) is a 2-simplicial neighbour, then the generalised binary matrix $\mathbf{X}^{(i,k)}$ is also firm.*

Proof. Let (ℓ, k) be the simplicial 1 of $\mathbf{X}$ which makes (i,k) a 2-simplicial neighbour.

Any submatrix of $\mathbf{X}^{(i,k)}$ that does not contain either of row i or row ℓ or column k is firm, because it is either a firm submatrix of $\mathbf{X}$ or a firm submatrix of $\mathbf{X}$ appended with a column that only contains 0s and a ?.

Let $\mathbf{Y}$ be an arbitrary submatrix of $\mathbf{X}$ indexed by $I \times J$ with $i, \ell \in I$ and $k \in J$. Let $\mathbf{Y}^{(i,k)}$ be the submatrix of $\mathbf{X}^{(i,k)}$ indexed by $I \times J$. Then we have $\mathfrak{i}(\mathbf{Y}^{(i,k)}) \leq \mathfrak{i}(\mathbf{Y})$ and $\mathfrak{br}(\mathbf{Y}^{(i,k)}) \leq \mathfrak{br}(\mathbf{Y})$. Since (ℓ,k) is a simplicial 1 of $\mathbf{Y}$, we may assume that it is the member of a maximum isolated set S of $\mathbf{Y}$. Then S is also a feasible isolated set for $\mathbf{Y}^{(i,k)}$, so $\mathfrak{i}(\mathbf{Y}) = \mathfrak{i}(\mathbf{Y}^{(i,k)})$. Then $\mathfrak{i}(\mathbf{Y}) = \mathfrak{i}(\mathbf{Y}^{(i,k)}) \leq \mathfrak{br}(\mathbf{Y}^{(i,k)}) \leq \mathfrak{br}(\mathbf{Y})$, and as $\mathbf{Y}$ is firm, we have $\mathfrak{i}(\mathbf{Y}^{(i,k)}) = \mathfrak{br}(\mathbf{Y}^{(i,k)})$. □

We say $\mathcal{S}^{(i,k)}(\mathbf{X})$ is obtained by *2-simplicial neighbour stretching* if (i,k) is a 2-simplicial neighbour. For instance, stretching applied to $(2,4)$ of $\mathbf{D}_4$ is an example of 2-simplicial neighbour stretching,

$$\mathcal{S}^{(2,4)}(\mathbf{D}_4) = \begin{bmatrix} 1 & 1 & & & \\ 1 & 1 & 1 & \colorbox{lightgray}{1} & 1 \\ & 1 & 1 & 1 & \\ & 1 & 1 & & \\ & & & 1 & 1 \end{bmatrix}.$$

In the next theorem, we show that 2-simplicial neighbour stretching preserves firmnesses and superfirmness, and that it can also be used to extend holes of rectangle cover graphs while maintaining their parity.

Theorem 4.3.9. *Let $\mathbf{X} \in \{0,1\}^{m \times n}$ have a simplicial 1 at position (ℓ,k), a 2-simplicial neighbour at (i,k), and let all other entries in column k be 0s.*

(1.) If $\mathbf{X}$ is superfirm, then so is $\mathcal{S}^{(i,k)}(\mathbf{X})$. In particular, 2-simplicial neighbour stretching preserves superfirmness.

(2.) If C is a hole in $\mathcal{G}(\mathbf{X})$ and $(i,j),(i,k),(\ell,t)$ are three vertices of C for some $t,j \in [n]$ then

$$C' = (C \setminus \{(i,k)\}) \cup \{(i,n+1),(m+1,k),(\ell,k)\}$$

is a hole of size $|C|+2$ in $\mathcal{G}(\mathcal{S}^{(i,k)}(\mathbf{X}))$.

(3.) If $\mathbf{X}$ is firm, then $\mathcal{S}^{(i,k)}(\mathbf{X})$ is also firm with $\mathfrak{i}(\mathcal{S}^{(i,k)}(\mathbf{X})) = \mathfrak{br}(\mathcal{S}^{(i,k)}(\mathbf{X})) = \mathfrak{br}(\mathbf{X})+1$. In particular, 2-simplicial neighbour stretching preserves firmness.

(4.) Let $\mathbf{X}$ have no repeated rows and no repeated columns and at least two 1s in each row and column.
If $\mathbf{X}$ has p simplicial 1s, then so has $\mathcal{S}^{(i,k)}(\mathbf{X})$. In particular, 2-simplicial neighbour stretching preserves the number of simplicial 1s of such matrices $\mathbf{X}$.

Proof. Let the unique maximal rectangle of the simplicial 1 at (ℓ,k) be given by $\{i,\ell\} \times (\{k\} \cup T)$ for some $T = \{t_1,\dots,t_{|T|}\}$, $|T| \geq 1$.

(1.) Let $\mathbf{X}$ be superfirm, and suppose that $\mathcal{S}^{(i,k)}(\mathbf{X})$ is not superfirm. Then by Theorem 4.1.3 the rectangle cover graph of $\mathcal{S}^{(i,k)}(\mathbf{X})$ contains an odd hole C' and C' is not in $\mathcal{G}(\mathbf{X})$.

The first two vertices that must be in C': Since simplicial 1s cannot be vertices of a hole, we must have $(i,n+1)$ and $(m+1,k)$ in C'.

The third vertex that must be in C': The only neighbours of $(i,n+1)$ in $\mathcal{G}(\mathbf{X})$ are in row i, while the only neighbours of $(m+1,k)$ are in column k. As (i,k) is adjacent to both $(i,n+1)$ and $(m+1,k)$, $(i,k) \notin C'$. Column k only has two 1s as (i,k) is 2-simplicial neighbour, so we must have $(\ell,k) \in C'$ as this is the only adjacent vertex to $(m+1,k)$ that is not adjacent to $(i,n+1)$.

The fourth vertex that must be in C': As $(i,n+1)$ is only adjacent to vertices in row i, and (ℓ,k) is adjacent to all vertices in the rectangle $\{i,\ell\} \times (\{k\} \cup T)$ and $|C'| \geq 5$, we must have $(i,j) \in C'$ for some column index j not in $\{k\} \cup T$.

So far we know four vertices that are in C', these are (i,j), $(i,n+1)$, $(m+1,k)$ and (ℓ,k). If $|C'| = 5$ then there is a vertex that is adjacent to both (i,j) and (ℓ,k) and not adjacent to $(i,n+1),(m+1,k)$. Looking at the below submatrix and remembering that (ℓ,k) is simplicial in $\mathbf{X}$, we conclude that such a vertex does not exist and $|C'| > 5$,

$$\begin{array}{c c} & \begin{array}{cccccc} j & t_1 & \cdots & t_{|T|} & k & n+1 \end{array} \\ \begin{array}{c} i \\ \ell \\ m+1 \end{array} & \left[\begin{array}{cccccc} \bullet & 1 & \dots & 1 & 1 & \bullet \\ & 1 & \dots & 1 & \bullet & \\ & & & & \bullet & 1 \end{array}\right]. \end{array}$$

The fifth vertex that must be in C': As (ℓ, k) is a simplicial 1 in $\mathbf{X}$, C' must have another vertex from row ℓ, say (ℓ, t) with $t \in T$. Therefore, the 5 vertices $(i, j), (i, n+1), (m+1, k), (\ell, k), (\ell, t)$ form a path and are all in C' and $|C'| > 5$. See this path on the left hand side of Figure 4.6.

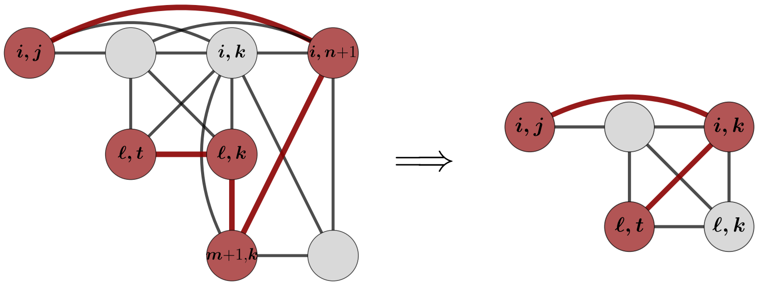


Figure 4.6: Decreasing the hole while keeping its parity

Consider $C = (C' \setminus \{(i, n+1), (m+1, k), (\ell, k)\}) \cup \{(i, k)\}$. Since (i, k) is 2-simplicial neighbour, it is only adjacent to (i, j) and (ℓ, t) in C, hence C is an odd hole of size $|C'| - 2$ in $\mathcal{G}(\mathbf{X})$. This contradicts the superfirmness of $\mathbf{X}$, so $\mathcal{S}^{(i,k)}(\mathbf{X})$ is superfirm.

(2.) To prove this part is essentially the backward argument of the previous part's proof. Let C be a hole in $\mathcal{G}(\mathbf{X})$ and $(i, j), (i, k), (\ell, t) \in C$ for some $t, j \in [n]$. As (ℓ, k) is a simplicial 1 and (i, k) is a 2-simplicial neighbour, we must have $t \in T$ and $j \notin T$, and the vertices $(i, j), (i, k), (\ell, t)$ must form a subgraph as shown on the right hand side of Figure 4.6. Consider $C' = (C \setminus \{(i, k)\}) \cup \{(i, n+1), (m+1, k), (\ell, k)\}$. Vertex $(i, n+1)$ is adjacent to only the vertices in row i from $\mathbf{X}$, (ℓ, k) is a simplicial vertex of $\mathcal{G}(\mathbf{X})$, $(m+1, k)$ is adjacent to only the vertices in column k from $\mathbf{X}$, and thus there is no chord between any of the vertices of C'. Therefore, C' is a hole of size $|C| + 2$ in $\mathcal{G}(\mathcal{S}^{(i,k)}(\mathbf{X}))$.

(3.) Let $\mathbf{Y}$ be an arbitrary submatrix of $\mathcal{S}^{(i,k)}(\mathbf{X})$ indexed by $I \times J$. If $m+1 \in I$ and $n+1 \in J$, then $(m+1, n+1)$ is a simplicial 1 of $\mathbf{Y}$ and we have $\mathfrak{i}(\mathbf{X}') + 1 = \mathfrak{i}(\mathbf{Y})$ and $\mathfrak{br}(\mathbf{X}') + 1 = \mathfrak{br}(\mathbf{Y})$ for some submatrix $\mathbf{X}'$ of the generalised binary matrix $\mathbf{X}^{(i,k)}$. By Lemma 4.3.8 $\mathbf{X}^{(i,k)}$ is firm, so $\mathbf{X}'$ is firm and $\mathfrak{br}(\mathbf{Y}) = \mathfrak{i}(\mathbf{Y})$.

If $m+1 \notin I$, then $\mathbf{Y}$ is either just a submatrix of $\mathbf{X}$ or $(i, n+1)$ is a simplicial 1 of $\mathbf{Y}$ and $\mathfrak{i}(\mathbf{Y}) = \mathfrak{i}(\mathbf{X}') + 1$ and $\mathfrak{br}(\mathbf{Y}) = \mathfrak{br}(\mathbf{X}') + 1$ for a firm submatrix $\mathbf{X}'$ of $\mathbf{X}$. The same holds if $n+1 \notin J$, so in any case $\mathfrak{i}(\mathbf{Y}) = \mathfrak{br}(\mathbf{Y})$.

(4.) Let $\mathbf{X}$ have no repeated rows and no repeated columns and at least two 1s in each row and column. Let $\mathbf{X}$ have p simplicial 1s. Observe that (ℓ, k) is not a simplicial 1 in $\mathcal{S}^{(i,k)}(\mathbf{X})$, but $(m+1, n+1)$ is.

We argue that no other simplicial 1s of $\mathbf{X}$ are affected by the stretching. Since $(i, n+1), (m+1, n+1), (m+1, k)$ are only adjacent to 1s in row i and column k, any simplicial 1 of $\mathbf{X}$ that is not in row i or column k remains a simplicial 1 in $\mathcal{S}^{(i,k)}(\mathbf{X})$. Since $\mathbf{X}$ has no repeated rows and column, only (ℓ, k) is a simplicial 1 in $\mathbf{X}$ from rectangle $\{(i,\ell)\} \times (\{k\} \cup T)$. Any other 1 in row i that is not in $\{(i,\ell)\} \times (\{k\} \cup T)$, say (i,j), cannot be simplicial because column j has at least two 1s, say another one at row i_1, and then $\{i_1, i\} \times \{j, k\}$ is a submatrix $\left[\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right]$, hence (i,j) is not simplicial. Therefore, $\mathcal{S}^{(i,k)}(\mathbf{X})$ has exactly $p-1+1$ simplicial 1s. $\square$

Let us see some examples why 2-simplicial neighbour stretching is interesting.

Example 4.3.10 (Extending holes and preserving parity)**.** *Recall that $\mathcal{G}(\mathbf{D}_3)$ has a 4-hole. The* 1 *at* $(2,3)$ *of* $\mathbf{D}_3$ *is a 2-simplicial neighbour as* $(3,3)$ *is a simplicial* 1*. $\mathcal{S}^{(2,3)}(\mathbf{D}_3)$ then contains a* 6*-hole as shown in Figure 4.7 and is superfirm by Theorem 4.3.9.*

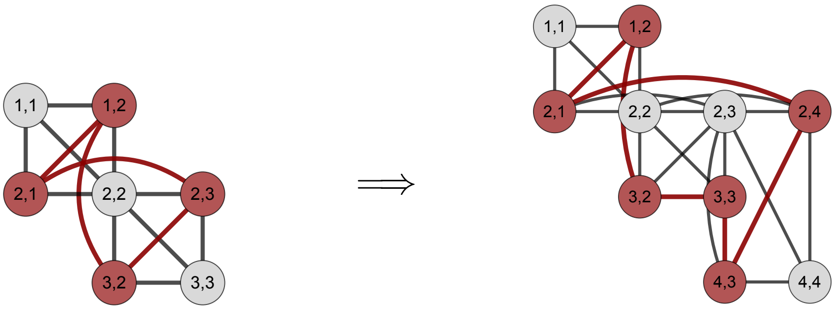


Figure 4.7: Creating a 6-hole from a 4-hole by stretching $(2,3)$

Recall that $\mathcal{G}(\mathbf{D}_4)$ contains a 5-hole. The 1 *at* $(2,4)$ *of* $\mathbf{D}_4$ *is a 2-simplicial neighbour as* $(3,4)$ *is a simplicial* 1 *with a maximal rectangle of size* 2×3*. Stretching* $(2,4)$ *we obtain the matrix shown on the right hand side of Figure 4.8. Observe that 2-simplicial neighbour stretching created a* 7*-hole and also preserved firmness.*

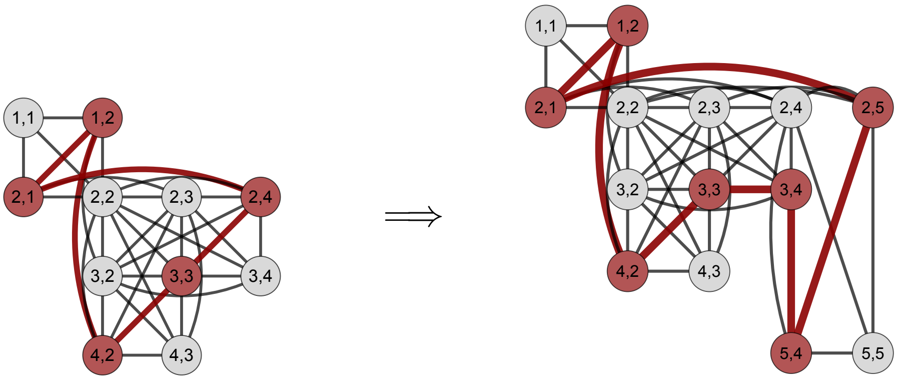


Figure 4.8: Creating a 7-hole from a 5-hole by stretching $(2, 4)$

Finally, one can show that stretching preserves one more property which is totally balancedness.

Lemma 4.3.11. *If* $\mathbf{X}$ *is totally balanced then so is* $\mathcal{S}^Q(\mathbf{X})$ *for any* $Q \subseteq \operatorname{supp}_1(\mathbf{X})$*. In particular, stretching preserves totally balancedness.*

Proof. Let $\mathbf{X}$ be a totally balanced matrix and let us assume that $\mathbf{X}$ is in a $\mathbf{\Gamma}$-free ordering. Let $(\ell, k) \in \operatorname{supp}_1(\mathbf{X})$. Consider an ordering of $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ in which the new row and column added by stretching are in the first row and column, so the three new 1s have positions $(1, 1), (1, k), (\ell, 1)$. Then this ordering is a $\mathbf{\Gamma}$-free ordering of $\mathcal{S}^{(\ell,k)}(\mathbf{X})$, hence $\mathcal{S}^{(\ell,k)}(\mathbf{X})$ is totally balanced by Theorem 2.3.1. □

4.4 Minimally non-superfirm matrices

Let us start our study of minimally non-superfirm matrices. Recall that a *minimally non-superfirm (mnsf)* matrix is not superfirm but all of its proper submatrices are. By Theorem 4.1.3 every non-superfirm matrix has an odd-hole and every mnsf matrix is mnsf due to the presence of an odd hole in the rectangle cover graph. Therefore, many results from Section 4.2 apply to this section. In particular, by Observation 4.2.1 we have the following simple but useful result.

Lemma 4.4.1. *A minimally non-superfirm matrix has at least two* 1*s in each row and column.*

Proof. Let $\mathbf{X}$ be minimally non-superfirm. Since $\mathbf{X}$ is not superfirm, by Theorem 4.1.3 $\mathcal{G}(\mathbf{X})$ contains an odd hole, and by minimality the odd hole has a vertex from each row and column of $\mathbf{X}$. Then Observation 4.2.1 5. applied to $\mathbf{X}$ and its odd hole tells us that $\mathbf{X}$ must have at least two 1s in each row and column. (Another way could be to see this is to observe that any single 1 in a row or column is simplicial and cannot be a vertex of a hole.) □

By Observation 4.2.1 we also know that the smallest dimension of an mnsf matrix containing a $2n-1$-hole is $n \times n$. The $n \times n$ matrices $\mathbf{M}_n$ $(n \geq 4)$,

$$\mathbf{M}_n = \mathcal{S}^{(n-1,n-1)}(\mathbf{C}_{n-1}) = \begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ 1 & & & & 1 & 1 \\ & & & & 1 & 1 \end{bmatrix},$$

that were first introduced by Lubiw [77] and observed not to be superfirm as $\mathcal{G}(\mathbf{M}_n)$ contains a $2n-1$-hole as shown in Figure 3.7 for $n = 4, 5$. Furthermore, it is easy to see that any proper submatrix of $\mathbf{M}_n$ is superfirm since if the submatrix has at least three rows or at least three columns then it is either $\mathbf{C}_{n-1}$ or has less than two 1s in a row or column. Therefore, $\mathbf{M}_n$ $(n \geq 4)$ is a class of mnsf matrices. In addition, since $\mathbf{M}_n$ is obtained by stretching the superfirm matrix $\mathbf{C}_{n-1}$, by Lemma 4.3.3, $\mathbf{M}_n$ is firm.

Lemma 4.4.2. $\mathbf{M}_n$ *is minimally non-superfirm and firm for all* $n \geq 4$.

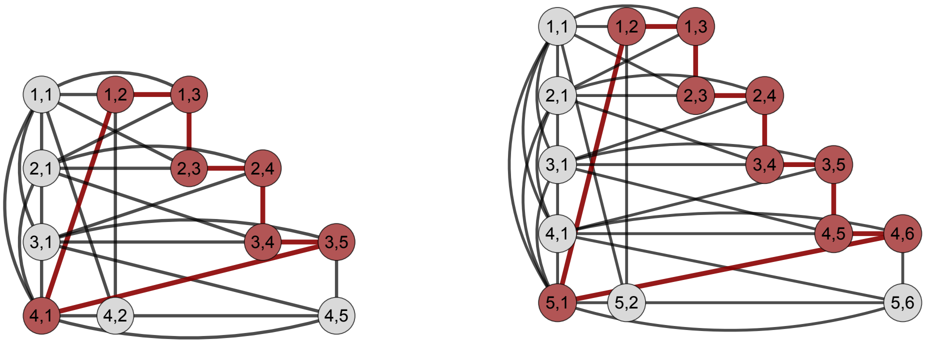
Recall that $\mathbf{H}_3$ is the 3×4 matrix whose appearance or its transpose's appearance is necessary for any odd antiholes of size 7 or more in rectangle cover graphs. Let us define for each $n \geq 3$,

$$\mathbf{H}_n := \begin{bmatrix} \mathbf{1} & \mathbf{C}_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & & & \\ 1 & & 1 & 1 & & \\ \vdots & & & \ddots & \ddots & \\ 1 & & & & 1 & 1 \\ 1 & 1 & & & & 1 \end{bmatrix}.$$

Similarly to $\mathbf{H}_3$, matrices $\mathbf{H}_n$ are not superfirm as their rectangle cover graph contains several odd holes, one of which is shown in Figure 4.9 for $n = 4, 5$. Next we show that $\mathbf{H}_n$ is mnsf and firm for all $n \geq 3$.

Lemma 4.4.3. $\mathbf{H}_n$ *is minimally non-superfirm and firm for* $n \geq 3$.

Proof. $\mathbf{H}_n$ is obtained by extending the superfirm matrix $\mathbf{C}_n$ by an all 1s column, hence it is firm by Lemma 3.1.13. $\mathcal{G}(\mathbf{H}_n)$ cannot have any odd-antiholes of size 7 or



(a) $\mathcal{G}(\mathbf{H}_4)$ (b) $\mathcal{G}(\mathbf{H}_5)$

Figure 4.9: The rectangle cover graph of $\mathbf{H}_4$ and $\mathbf{H}_5$ with one of their odd holes highlighted

more as it does not have any of the two 4×4 matrices from Lemma 4.1.2. On the other hand, $\mathcal{G}(\mathbf{H}_n)$ contains n $2n-1$-holes given by

$$\begin{aligned} C^i &= \operatorname{supp}_1(\mathbf{H}_n) \setminus \Big(\{(\ell,1) : \ell \neq i\} \cup \{(i,i+1),(i,i+2)\}\Big) && \text{for } i \in [n-1], \\ C^i &= \operatorname{supp}_1(\mathbf{H}_n) \setminus \Big(\{(\ell,1) : \ell \neq i\} \cup \{(i,i+1),(i,2)\}\Big) && \text{for } i = n. \end{aligned}$$

Any other hole in $\mathcal{G}(\mathbf{H}_n)$ is either contained in the submatrix $\mathbf{C}_n$ and hence it is the $2n$-hole, or contains at most two vertices from column 1. Note that if $(\ell, 1)$ is a vertex of a hole then the hole cannot have another vertex from row ℓ. If a hole contains a single vertex from column 1 then it is one of the n $2n-1$-holes. If the hole has two vertices from column 1, then it must contain an even number of vertices from submatrix $\mathbf{C}_n$, so it is an even hole. Therefore, the n $2n-1$-holes are the only odd holes in $\mathcal{G}(\mathbf{H}_n)$ which all have a vertex from each row and column of $\mathbf{H}_n$ and thus no proper submatrix can contain them. Hence, every proper submatrix of $\mathbf{H}_n$ is superfirm and $\mathbf{H}_n$ is mnsf. □

Recall the interval matrix $\mathbf{D}_4$ from Equation (3.1.1).

Lemma 4.4.4. $\mathbf{D}_4$ *is minimally non-superfirm and firm with* $\mathfrak{i}(\mathbf{D}_4) = \mathfrak{br}(\mathbf{D}_4) = 3$.

Proof. $\mathbf{D}_4$ has three simplicial 1s at $(1,1)$, $(3,4)$ and $(4,3)$, and these form an isolated set of size 3. On the other hand, the three unique maximal rectangles corresponding to these three simplicial 1s form a feasible rectangle cover for $\mathbf{D}_4$. Hence we have $\mathfrak{i}(\mathbf{D}_4) = \mathfrak{br}(\mathbf{D}_4) = 3$.

$\mathcal{G}(\mathbf{D}_4)$ contains the 5-hole shown in Figure 3.2 which has a vertex from each row and column. Since $\mathbf{D}_4$ is 4×4 is cannot contain an odd-hole of size 7 or more, and by

Lemma 4.1.2 it clearly cannot have an odd antihole of size 7 or more. Furthermore, by Theorem 4.2.3 no proper submatrix of it can contain any 5-holes and thus every proper submatrix of $\mathbf{D}_4$ is superfirm. $\square$

In computational experiments, we enumerated several interval matrices. On every example we considered, we observed that if an interval matrix has an odd hole in its rectangle cover graph then it also contains a $\mathbf{D}_4$ submatrix. Hence we conjecture that $\mathbf{D}_4$ is the only minimally non-superfirm interval matrix. While we cannot prove this result, a simple lemma that is useful to know about odd holes in rectangle cover graphs is proved in Appendix A.1.

Let $\mathbf{T}_5$ be the matrix obtained by stretching two 1s of $\mathbf{D}_3$,

$$\mathcal{S}^{\{(2,3),(3,2)\}}(\mathbf{D}_3)) = \begin{bmatrix} 1 & 1 & & & \\ 1 & 1 & 1 & 1 & \\ & 1 & 1 & & 1 \\ & & 1 & 1 & \\ & 1 & & & 1 \end{bmatrix}.$$

$\mathbf{T}_5$ has a 7-hole in its rectangle cover graph as shown on the right hand side of Figure 4.4. Observe that $\mathbf{T}_5$ can be permuted to be symmetric and from now on we will only use this symmetric ordering for $\mathbf{T}_5$,

$$\mathbf{T}_5 = \begin{bmatrix} 1 & 1 & & & \\ 1 & 1 & 1 & & 1 \\ & 1 & 1 & 1 & \\ & & 1 & & 1 \\ & 1 & & 1 & \end{bmatrix}.$$

Lemma 4.4.5. $\mathbf{T}_5$ *is totally balanced, minimally non-superfirm and firm with* $\mathfrak{i}(\mathbf{T}_5) = \mathfrak{br}(\mathbf{T}_5) = 4$.

Proof. Since stretching preserves totally balancedness $\mathbf{T}_5$ is totally balanced and cannot have any odd antiholes of size 7 or more by Lemma 4.1.2.

Note that $\mathbf{T}_5$ does not have a $\mathbf{D}_4$ submatrix and $\mathbf{D}_4$ is the only totally balanced matrix responsible for the appearance of 5-holes, so $\mathcal{G}(\mathbf{T}_5)$ does not have any 5-holes. Any potential non-superfirm proper submatrix of $\mathbf{T}_5$ thus contains a 7-hole and must be of dimension at least 4×4, the minimum dimension needed for a 7-hole. Since three rows and three columns of $\mathbf{T}_5$ contain exactly two 1s, the only proper submatrices that need to be checked for superfirmness are $\{1,2,3,4\} \times \{1,2,3,5\}$ and $\{2,3,4,5\} \times \{2,3,4,5\}$, the rest of the 4×4 matrices with at least two 1s in each row and column are permutations of these. These two matrices can be seen to be superfirm as one of them is $\mathcal{S}^{(2,3)}(\mathbf{D}_3)$ and $(2,3)$ of $\mathbf{D}_3$ is 2-simplicial neighbour, and the other is just $\mathcal{S}^{(3,3)}(\mathbf{D}_3)$ which is easily seen to be superfirm. Therefore $\mathbf{T}_5$ is mnsf.

Furthermore, $\mathbf{T}_5$ is firm because $\{(1,1),(3,3),(4,5),(5,4)\}$ is an isolated set of size 4 and the four rectangles $\{1,2\}\times\{1,2\}$,$\{2,3\}\times\{2,3\}$, $\{3,5\}\times\{2,4\}$, $\{2,4\}\times\{3,5\}$ cover it. $\square$

Let $\mathbf{W}_n$ for $n \geq 5$, be the matrices defined below,

$$\mathbf{W}_5 = \begin{bmatrix} 1 & 1 & & & \\ 1 & & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & \\ & 1 & 1 & & \end{bmatrix}, \qquad \mathbf{W}_n = \begin{bmatrix} 1 & 1 & & & & & \\ 1 & & \ddots & & & & \\ & \ddots & & 1 & & & \\ & & 1 & & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & \\ & & & 1 & 1 & & \end{bmatrix}.$$

The rectangle cover graph of $\mathbf{W}_n$ contains a $2n-3$-hole as shown in Figure 4.10 for $n = 5$. Next we show that these matrices are also mnsf and firm.

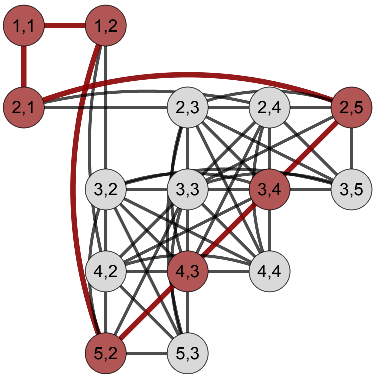


Figure 4.10: The rectangle cover graph of $\mathbf{W}_5$ with its 7-hole highlighted

Lemma 4.4.6. $\mathbf{W}_n$ *is minimally non-superfirm and firm for all* $n \geq 5$.

Proof. For n odd or even let S_n be

$$S_{2k+1} = \{(1,2),(3,4),\ldots,(n-4),(n-3)\} \cup \{(2,1),(4,3),\ldots,(n-3,n-4)\},$$
$$S_{2k} = \{(1,1)\} \cup \{(2,3),(4,5),\ldots,(n-4),(n-3)\} \cup \{(3,2),(5,4),\ldots,(n-3,n-4)\}.$$

Then $S = S_n \cup \{(n,n-2),(n-1,n-1),(n-2,n)\}$ gives an isolated set of size n, hence $\mathfrak{i}(\mathbf{W}_n) = \mathfrak{br}(\mathbf{W}_n) = n$ for all $n \geq 5$.

Suppose that $\mathbf{W}_n$ is not mnsf and has a proper submatrix $\mathbf{Y}$ indexed by $I \times J$ which is mnsf. Then by Lemma 4.4.1 $\mathbf{Y}$ has at least two 1s in each row and column. If

matrix	dimension	$\mathfrak{i}$	$\mathfrak{br}$	$\lvert C\rvert$	$n_d(C)$	$n_v(C)$	$n_h(C)$
$\mathbf{D}_4$	4×4	3	3	5	3	1	1
$\mathbf{T}_5$	5×5	4	4	7	3	2	2
$\mathbf{M}_n, n\geq 4$	$n\times n$	n	n	$2n-1$	1	$n-1$	$n-1$
$\mathbf{H}_n$, $n\geq 3$	$n\times(n+1)$	n	n	$2n-1$	2	$n-2$	$n-1$
$\mathbf{W}_n, n\geq 5$	$n\times n$	n	n	$2n-3$	3	$n-3$	$n-3$

Table 4.1: A list of characteristics of so far known mnsf matrices

$I\subseteq[n-3,n]$ or $J\subseteq[n-3,n]$ then it is easy to see that $\mathbf{Y}$ is superfirm. Therefore, $\mathbf{Y}$ must have at least one row or column with an index in $[n-4]$. Any row or column with index in $[n-4]$ has exactly two 1s, so I and J must contain the indices of both of those 1s. Therefore $[n-3]\subset I$ and $[n-3]\subset J$. Submatrix $[n-2]\times[n-2]$ is $\mathbf{C}_{n-2}$ which is superfirm, so we must have $|I\cap\{n-2,n-1,n\}|=2$ and $|J\cap\{n-2,n-1,n\}|=2$ or $\mathbf{Y}$ is either just $\mathbf{C}_{n-2}$ or $\mathbf{C}_{n-2}$ with repeated rows or columns. Hence, $\mathbf{Y}$ must be as shown below,

$$\mathbf{Y}=\begin{bmatrix} 1 & 1 & & & & \\ 1 & & \ddots & & & \\ & \ddots & & 1 & & \\ & & 1 & & 1 & 1 \\ & & & 1 & 1 & 1 \\ & & & 1 & 1 & \end{bmatrix}.$$

However, then $\mathbf{Y}$ is easily seen to be superfirm as it only contains $2n-4$-holes. □

Table 4.1 gives a summary of the mnsf matrices that we considered in this section. C denotes the odd hole that is in the mnsf matrix and $n_d(C), n_v(C), n_h(C)$ denote the number of diagonal, vertical and horizontal edges of C, respectively.

Observe that all mnsf matrices in Table 4.1 have dimension $m\times n$ satisfying $|m-n|\leq 1$ and the square mnsf matrices can all be permuted to a symmetric matrix form. In addition, all mnsf matrices are firm. We are curious whether these observations hold for all mnsf matrices.

Lemma 4.4.7. *If a minimally non-superfirm matrix has an all* 1*s row or column, or a simplicial* 1 *then it is firm.*

Proof. Let $\mathbf{X}$ be mnsf. If $\mathbf{X}$ has an all 1s row, then $\mathbf{X}$ is obtained by appending a superfirm matrix with an all 1s row and hence firm by Lemma 3.1.13.

If $\mathbf{X}$ has a simplicial 1 at (ℓ,k), then by removing that simplicial 1, we delete row ℓ and column k and turn the 1's that are in (ℓ,l)'s maximal rectangle into ?'s. Let these 1's be indexed by set K. Then the matrix we obtain is $\mathbf{Y}^K$, where $\mathbf{Y}$ a proper submatrix of $\mathbf{X}$ and thus superfirm. By Lemma 4.3.1, we have $\mathfrak{i}(\mathbf{X}) = \mathfrak{i}(\mathbf{Y}^K) + 1 = \mathfrak{br}(\mathbf{Y}^K) + 1 = \mathfrak{br}(\mathbf{X})$, thus $\mathbf{X}$ is firm. □

Recall that every totally balanced matrix has at least one simplicial 1 which can be found by computing a $\mathbf{\Gamma}$-free ordering of $\mathbf{X}$. Therefore we obtain the following corollary of Lemma 4.4.7.

Corollary 4.4.8. *Every totally balanced minimally non-superfirm matrix is firm.*

We suspect that not only totally balanced mnsf matrices are firm but every mnsf matrix. We however cannot prove this so we state this as a conjecture.

Conjecture 4.4.9. *Every minimally non-superfirm matrix is firm and is of dimension $m \times n$ with $|m - n| \leq 1$.*

Chapter 5

Minimally non-firm matrices

A binary matrix is said to be *minimally non-firm (mnf)* if it is not firm but all of its proper submatrices are. In this chapter, we first explore a few general properties of mnf matrices and then construct several infinite families of them. We will show that for every minimally non-superfirm matrix considered in the previous chapter there is at least one mnf matrix containing it. We will use the stretching operation's firmness preserving and firmness destroying properties to construct the mnf matrices.

To the best of our knowledge, mnf binary matrices have not been explicitly studied before. However, there are a few mnf matrices that are mentioned in the works of Lubiw [77] and Caen et al. [28] with only observing their non-firmness and not their *minimally* non-firmness. The first matrix that was showed to be non-firm is $\overline{\mathbf{I}}_4 = \mathbf{J}_4 - \mathbf{I}_4$, the complement identity matrix of size 4. Recall that in Example 1.2.1 it is showed that $\mathfrak{i}(\overline{\mathbf{I}}_4) = 3 < \mathfrak{br}(\overline{\mathbf{I}}_4) = 4$. The second well known non-firm matrix is the totally balanced swath matrix of Chung's polygon which is given in Equation (2.1.1) and the minimally non-firm matrix obtained from it by removing some rows and columns that is given in Equation (2.1.2). The mnf matrix from Chung's polygon was first presented by Lubiw [77, Fig 1.1] and observed to be non-firm but she did not mention it to be minimally non-firm. She did present another mnf matrix in [77, Fig 1.1], but again just to emphasise that the Boolean rank is not always equal to the isolation number. These two mnf matrices that Lubiw presented, inspired most of the work in this chapter. Both of these mnf matrices can be obtained by a series of stretchings from the mnsf matrix $\mathbf{D}_4$, and while trying to prove and understand why they are mnf, we came up with a generalisation idea to show other matrices to be mnf.

5.1 Preliminaries

Let us start by immediately extending the definition of minimally non-firmness to generalised binary matrices. Recall from the introduction that a matrix $\mathbf{X}$ over $\{0,1,?\}$ is a generalised binary matrix and ?'s can be used to form rectangles, cannot be in isolated sets and need not be covered in a rectangle covering. A generalised binary matrix $\mathbf{X}$ is *mnf* if $\mathfrak{i}(\mathbf{X}) < \mathfrak{br}(\mathbf{X})$ and $\mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}')$ for all proper submatrices $\mathbf{X}'$ of $\mathbf{X}$.

Given a generalised binary matrix $\mathbf{X}$, if $\mathbf{X}'$ is obtained from $\mathbf{X}$ by deleting a single row or column then we clearly have

$$\mathfrak{i}(\mathbf{X}) - 1 \leq \mathfrak{i}(\mathbf{X}') \leq \mathfrak{i}(\mathbf{X}) \qquad \text{and} \qquad \mathfrak{br}(\mathbf{X}) - 1 \leq \mathfrak{br}(\mathbf{X}') \leq \mathfrak{br}(\mathbf{X}), \tag{5.1.1}$$

as a single row or column is a rectangle and may contain only one element from an isolated set. Now if $\mathbf{X}$ is mnf and $\mathbf{X}'$ is obtained from $\mathbf{X}$ by dropping a single row or column, then $\mathbf{X}'$ is firm and satisfies Equation (5.1.1), so it must have

$$\mathfrak{br}(\mathbf{X}) - 1 \leq \mathfrak{i}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}') \leq \mathfrak{i}(\mathbf{X}) < \mathfrak{br}(\mathbf{X}).$$

Therefore, we have the following two observations which apply to both standard binary and generalised binary mnf matrices.

Observation 5.1.1. *If $\mathbf{X} \in \{0,1,?\}^{m \times n}$ is mnf and $\mathbf{X}'$ is obtained from $\mathbf{X}$ by removing a single row or column then $\mathfrak{i}(\mathbf{X}') = \mathfrak{i}(\mathbf{X})$ and $\mathfrak{br}(\mathbf{X}') = \mathfrak{br}(\mathbf{X}) - 1$.*

Observation 5.1.2. *If $\mathbf{X} \in \{0,1,?\}^{m \times n}$ is mnf then $\mathfrak{i}(\mathbf{X}) = \mathfrak{br}(\mathbf{X}) - 1$.*

Recall that if a 1 is the single non-zero entry in a row or column of a generalised binary matrix $\mathbf{X}$ then it is a simplicial 1. By Lemma 4.3.1, the simplicial 1 can be removed to obtain a proper submatrix $\mathbf{X}'$ of $\mathbf{X}$ which satisfies $\mathfrak{i}(\mathbf{X}') + 1 = \mathfrak{i}(\mathbf{X})$ and $\mathfrak{br}(\mathbf{X}') + 1 = \mathfrak{br}(\mathbf{X})$. Clearly a 0 row or column, or a row or column which only contains ?'s can be dropped without changing the isolation number and Boolean rank, hence we get the next lemma about mnf matrices.

Lemma 5.1.3. *If $\mathbf{X} \in \{0,1,?\}^{m \times n}$ is mnf, then every row and column of $\mathbf{X}$ contains at least two non-zero entries, and at least one of these non-zero entries is a 1.*

If $\mathbf{X}$ is a minimally non-firm standard or generalised binary matrix then it must be impossible to decompose it into any of its proper submatrices via any firmness preserving operation. Recall that in Section 3.3.2 we present Lubiw's L-sum operation that preserves firmness and also superfirmness. Since any matrix that can be L-decomposed is decomposed into the L-sum of two of its proper submatrices, minimally non-firm matrices cannot be L-decomposed. The reasoning holds for the direct sum operation which preserves firmness, as clearly an mnf matrix cannot have a block diagonal structure.

We have also seen some operations related to Boolean linear algebra that preserve the isolation number and Boolean rank, so let us recall some Boolean linear algebra definitions from Section 1.1.1. For a binary matrix $\mathbf{X}$, $BRS(\mathbf{X})$ and $BCS(\mathbf{X})$ are the Boolean row and column spaces which contain all binary vectors that can be expressed as the Boolean sum of rows and columns of $\mathbf{X}$ using binary coefficients, respectively. The Boolean row (or column) rank of $\mathbf{X}$ is the cardinality of the set of rows (or columns) of $\mathbf{X}$ that form a maximum Boolean independent set in $BRS(\mathbf{X})$ (or $BCS(\mathbf{X})$), where a set of binary vectors is Boolean independent if no vector of the set can be expressed as the Boolean sum of some other vectors in the set.

Lemma 3.1.11 says that whenever a binary matrix $\mathbf{X}$ is appended by a row vector from $BRS(\mathbf{X})$ then the isolation number and Boolean rank remain the same. This implies that if $\mathbf{X}$ has a row which is the Boolean sum of some other rows, then dropping that row does not change the isolation number nor the Boolean rank. By Observation 5.1.1, if $\mathbf{X}$ is mnf, then dropping any row or column of $\mathbf{X}$ must reduce its Boolean rank. Therefore, an mnf matrix cannot have a row or column that is the Boolean sum of some other rows or columns. This argument shows that the rows and columns of any standard binary mnf matrix must form a Boolean independent set and we have just proved the following lemma.

Lemma 5.1.4. *If $\mathbf{X} \in \{0,1\}^{m \times n}$ is minimally non-firm then $\mathbf{X}$ has Boolean row rank m and Boolean column rank n.*

We are interested in understanding what are the possible dimensions for mnf matrices. First, we thought that they may all need to be square matrices and then found some minimally non-firm matrices with dimension $(n-1) \times n$. Now we think that the following conjecture holds.

Conjecture 5.1.5. *If $\mathbf{X} \in \{0,1\}^{m \times n}$ is minimally non-firm, then $|m-n| \leq 1$.*

Our attempts to prove the above so far have been unsuccessful. We suspect that this dimension conjecture is intimately related to the same dimension conjecture about minimally non-superfirm matrices that is stated in Conjecture 4.4.9.

5.2 The smallest minimally non-firm matrices

What is the dimension of the smallest minimally non-firm standard binary matrix? Any matrix with only two rows or two columns is superfirm, and any matrix that has only three rows or three columns can only have a 5-hole in it if it contains $\mathbf{H}_3$ or its transpose as a submatrix by Theorem 4.2.3. Since $\mathbf{H}_3$ is firm, we get the following corollary of Theorem 4.2.3.

Corollary 5.2.1. *A binary matrix with only three rows or only three columns is firm.*

Using corollary 5.2.1 we can prove the minimally non-firmness of the identity complement matrix of size 4 × 4. As shown in Example 1.2.1, $\mathfrak{i}(\overline{\mathbf{I}}_4) = 3 < \mathfrak{br}(\overline{\mathbf{I}}_4) = 4$ and all proper submatrices of $\overline{\mathbf{I}}_4$ have only three rows or three columns hence $\overline{\mathbf{I}}_4$ is mnf.

Is there any other mnf matrix of dimension 4 × 4? Let $\overline{\mathbf{I}}'_4$ be obtained by turning an arbitrary 1 of $\overline{\mathbf{I}}_4$ to a 0,

$$\overline{\mathbf{I}}'_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

In the next lemma, we show that $\overline{\mathbf{I}}'_4$ is also minimally non-firm.

Lemma 5.2.2. $\overline{\mathbf{I}}'_4$ *is minimally non-firm.*

Proof. For a matrix to have an isolated set of cardinality k, it needs to have at least $\binom{k}{2}$ 0s by Equation (1.2.2). Since $\overline{\mathbf{I}}'_4$ only has five 0s, $\mathfrak{i}(\overline{\mathbf{I}}'_4) \leq 3$. On the other hand, $\overline{\mathbf{I}}'_4$ contains a submatrix $\mathbf{C}_3$ so $\mathfrak{i}(\overline{\mathbf{I}}'_4) = 3$. To see that $\mathfrak{br}(\overline{\mathbf{I}}'_4) = 4$, observe that the generalised binary matrix,

$$\overline{\mathbf{I}}'^P_4 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ ? & 0 & 1 & 1 \\ ? & 1 & 0 & 1 \\ 1 & ? & ? & 0 \end{bmatrix},$$

has 7 1s and largest rectangle of size 2. Hence $\lceil \frac{7}{2} \rceil \leq \mathfrak{br}(\overline{\mathbf{I}}'^P_4) \leq \mathfrak{br}(\overline{\mathbf{I}}'_4)$ by Equation (1.3.2). Furthermore, any proper submatrix of $\overline{\mathbf{I}}'_4$ has only three rows or three columns, hence by Corollary 5.2.1 it is firm and $\overline{\mathbf{I}}'_4$ is mnf. □

Is there another minimally non-firm standard binary matrix of dimension 4 × 4 different from $\overline{\mathbf{I}}_4$ and $\overline{\mathbf{I}}'_4$? We show in the next lemma that there is no such matrix.

Lemma 5.2.3. $\overline{\mathbf{I}}_4$ *and* $\overline{\mathbf{I}}'_4$ *are the only minimally non-firm binary matrices of dimension* 4 × 4.

Proof. Let $\mathbf{X} \in \{0,1\}^{4\times 4}$ be mnf that is not $\overline{\mathbf{I}}_4$ and $\overline{\mathbf{I}}_4'$. Then $\mathbf{X}$ cannot have duplicated rows or columns or an all 1s row or column because then it is obtained form a smaller matrix by a firmness preserving operation and hence firm. With dimension 4×4, $\mathcal{G}(\mathbf{X})$ can contain a 5- or a 7-hole.

(a) If it contains a 5-hole it must have a $\mathbf{H}_3$ or $\mathbf{H}_3^\top$ submatrix by Theorem 4.2.3 (as $\mathbf{K}_5$ is too large, and $\mathbf{D}_4$ is 4×4 but firm). So let $\mathbf{X}$ contain $\mathbf{H}_3$ (or $\mathbf{H}_3^\top$, which is a symmetric case) as the submatrix formed by rows $\{2,3,4\}$. Then only the first row of $\mathbf{X}$ is not determined yet. As column 1 so far has only 1s, we must have $x_{11} = 0$. Then if $x_{12} = x_{13} = x_{14} = 1$, $\mathbf{X} = \overline{\mathbf{I}}_4$, and if two of these entries are 1s then $\mathbf{X} = \overline{\mathbf{I}}_4'$. Hence the first row of $\mathbf{X}$ is a singleton or a zero row and $\mathbf{X}$ is firm.

(b) If $\mathcal{G}(\mathbf{X})$ contains a 7-hole then it has at least 9 1s, at least two 1s in each row and column. Then three rows and three columns contain exactly 2 vertices of C_7 and exactly one row and one column contain only one vertex of C_7, and C_7 must have 3 vertical, 3 horizontal and 1 diagonal edges which shows that $\mathbf{M}_4 \leq \mathbf{X}$. If $\mathbf{X}$ has a tenth 1 then a chord appears in C_7, therefore we must have $\mathbf{X} = \mathbf{M}_4$ which is a firm matrix.

Therefore, $\overline{\mathbf{I}}_4$ and $\overline{\mathbf{I}}_4'$ are the only minimally non-firm 4×4 matrices. □

5.3 Minimally non-firm matrices from generalised binary matrices

In this section, we prove a theorem which will be used in the rest of the chapter to create minimally non-firm binary matrices by stretching a carefully selected subset of 1s of some matrices that have odd holes in their rectangle cover graphs.

By Theorem 4.1.3, any matrix is superfirm if it has no odd holes in its rectangle cover graph, so for a matrix to be mnf, its rectangle cover graph must contain an odd hole. Recall that for some non-empty set $P \subset \mathrm{supp}_1(\mathbf{X})$, $\mathbf{X}^P$ denotes the generalised binary matrix that is obtained from $\mathbf{X}$ by replacing the 1s at P by ?s. Let $\mathbf{X}$ be a standard binary matrix with an odd hole C in $\mathcal{G}(\mathbf{X})$ of size $|C| = 2k+1$ and let $Q = \mathrm{supp}_1(\mathbf{X}) \setminus C$. By stretching the 1s at Q of $\mathbf{X}$, we obtain $\mathcal{S}^Q(\mathbf{X})$ which has $|Q|$ simplicial 1s that are created by the stretchings. Removing these $|Q|$ simplicial 1s from $\mathcal{S}^Q(\mathbf{X})$ and applying Lemma 4.3.1, we 'expose' the odd hole in the rectangle cover graph of $\mathbf{X}$:

$$\mathfrak{i}(\mathcal{S}^Q(\mathbf{X})) - |Q| = \mathfrak{i}(\mathbf{X}^Q) = k < k+1 = \mathfrak{br}(\mathbf{X}^Q) = \mathfrak{br}(\mathcal{S}^Q(\mathbf{X})) - |Q|.$$

Therefore, $\mathcal{S}^Q(\mathbf{X})$ is a non-firm matrix. This selection of $Q \subset \operatorname{supp}_1(\mathbf{X})$ however, does not guarantee that $\mathcal{S}^Q(\mathbf{X})$ is *minimally* non-firm. By adding extra conditions on Q, minimality can be enforced.

Theorem 5.3.1. *Let $\mathbf{X} \in \{0,1\}^{m\times n}$. If $\mathbf{X}^Q$ is a minimally non-firm generalised binary matrix for some non-empty $Q \subset \operatorname{supp}_1(\mathbf{X})$ and $\mathbf{X}^P$ is firm for all $P \subsetneq Q$, then $\mathcal{S}^Q(\mathbf{X}) \in \{0,1\}^{(m+|Q|)\times(n+|Q|)}$ is minimally non-firm.*

Proof. $\mathcal{S}^Q(\mathbf{X})$ may be written as a block matrix with four blocks $\mathbf{X}, \mathbf{L}, \mathbf{U}$ and $\mathbf{I}_{|Q|}$ as in Equation (4.3.1),

$$\mathcal{S}^Q(\mathbf{X}) = \begin{bmatrix} \mathbf{X} & \mathbf{U} \\ \mathbf{L} & \mathbf{I}_{|Q|} \end{bmatrix},$$

where $\mathbf{U}$ is $m \times |Q|$ and has exactly one 1 in each column, and $\mathbf{L}$ is $|Q| \times n$ with exactly one 1 in each row. By construction all 1s in block $\mathbf{I}_{|Q|}$ are simplicial, hence removing them we obtain the mnf generalised binary matrix $\mathbf{X}^Q$. By Lemma 4.3.1 then $\mathfrak{i}(\mathcal{S}^Q(\mathbf{X})) = \mathfrak{i}(\mathbf{X}^Q) + |Q| < \mathfrak{br}(\mathbf{X}^Q) + |Q| = \mathfrak{br}(\mathcal{S}^Q(\mathbf{X}))$.

Suppose that not all proper submatrices of $\mathcal{S}^Q(\mathbf{X})$ are firm and let $\mathbf{Y}$ be a smallest non-firm proper submatrix of $\mathcal{S}^Q(\mathbf{X})$ indexed by $I \times J$. Then $\mathbf{Y}$ is mnf. Note that the four block matrices of $\mathcal{S}^Q(\mathbf{X})$ are all firm: (1) $\mathbf{X}$ is firm as it is just $\mathbf{X}^{\emptyset}$. (2) $\mathbf{I}_{|Q|}$ is clearly firm. (3) $\mathbf{U}$ has exactly one 1 per column, so it can be obtained from an identity matrix by duplicating columns and adding zero rows, and thus firm. (4) Similarly, as $\mathbf{L}$ has exactly one 1 per row, it is firm. Hence $\mathbf{Y}$ cannot be fully contained in any of the four blocks. As $\mathbf{Y}$ is a mnf standard binary matrix it has at least two 1s in each row and column by Lemma 5.1.3. Since block $[\, \mathbf{L}\ \mathbf{I}_{|Q|} \,]$ has exactly two 1s in each row, if $\mathbf{Y}$ has a row from this block, then $\mathbf{Y}$ must also contain the columns of both 1s in this row. Similarly, if $\mathbf{Y}$ contains a column from block $\left[\begin{smallmatrix} \mathbf{U} \\ \mathbf{I}_{|Q|} \end{smallmatrix}\right]$, it must contain the rows of both 1s in this column. Therefore, the rows in I from block $[\, \mathbf{L}\ \mathbf{I}_{|Q|} \,]$ and the columns in J from block $\left[\begin{smallmatrix} \mathbf{U} \\ \mathbf{I}_{|Q|} \end{smallmatrix}\right]$ come in pairs and may be identified with their 1 in block $\mathbf{I}_{|Q|}$. Let P be the subset of Q whose stretching created the 1s in block $\mathbf{I}_{|Q|}$ which are in $\mathbf{Y}$. Removing all $|P|$ simplicial 1s present in $\mathbf{Y}$ from block $\mathbf{I}_{|Q|}$ we obtain a generalised binary matrix which is fully contained in block $\mathbf{X}$ and is just a submatrix $\mathbf{Z}$ of $\mathbf{X}^P$. By Lemma 4.3.1, $\mathbf{Z}$ satisfies $\mathfrak{i}(\mathbf{Z}) + |P| = \mathfrak{i}(\mathbf{Y})$ and $\mathfrak{br}(\mathbf{Z}) + |P| = \mathfrak{br}(\mathbf{Y})$. If $P = Q$, then $I \times J$ contains all the rows and columns from block $\mathbf{I}_{|Q|}$ so $\mathbf{Z}$ must be a proper submatrix of $\mathbf{X}^Q$, hence firm. If $P \neq Q$, then $\mathbf{Z}$ is a submatrix of the firm matrix $\mathbf{X}^P$. In both cases $\mathfrak{i}(\mathbf{Z}) = \mathfrak{br}(\mathbf{Z})$ which implies $\mathfrak{i}(\mathbf{Y}) = \mathfrak{br}(\mathbf{Y})$, a contradiction. □

This theorem provides a general 'recipe' on how to create mnf standard binary matrices from matrices that have some odd-holes in their rectangle cover graph and have a subset Q of 1s that satisfy all the conditions. Of course, the difficulty is to choose Q carefully and in the following sections we will show how to choose it for some classes of matrices.

Note that not all mnf standard binary matrices are obtained via stretching, for instance $\overline{\mathbf{I}}_4$ and $\overline{\mathbf{I}}'_4$ do not have any simplicial 1s so are not created by stretching.

On the other hand, if an mnf standard binary matrix has some simplicial 1s then removing those simplicial 1s gives an mnf generalised binary matrix. This observation gives a partial reverse of Theorem 5.3.1. We would be interested if a complete reverse of Theorem 5.3.1 is also true.

Conjecture 5.3.2. *If a minimally non-firm matrix has some simplicial* 1*s, then it is created by stretching.*

To prove this conjecture we would need to show that the unique maximal rectangle of at least one simplicial 1 in such mnf matrices is of size 2×2. This is because by stretching we can only create simplicial 1s whose unique maximal rectangles are of size 2×2.

5.4 Mnf matrices from mnsf matrices

In this section, we create a minimally non-firm matrix from each minimally non-superfirm matrix introduced in the previous chapter. We do this by identifying a subset Q of the 1s of the mnsf matrix and then show that Q satisfies the conditions of Theorem 5.3.1.

Let us start by a general theorem that we can then apply to mnsf matrices $\mathbf{D}_4, \mathbf{T}_5, \mathbf{H}_n, \mathbf{M}_n$ to obtain mnf matrices from them.

Theorem 5.4.1. *Let* $\mathbf{X}$ *be a minimally non-superfirm binary matrix with* $\mathfrak{br}(\mathbf{X}) = k$ *for some* $k \geq 3$*. If* $\operatorname{supp}_1(\mathbf{X})$ *can be partitioned into three sets* C*,* K *and* Q*, such that*

- *C induces a $2k-1$ hole in $\mathcal{G}(\mathbf{X})$,*
- *Q is the set of all vertices that are adjacent in $\mathcal{G}(\mathbf{X})$ to exactly two vertices of C and those two vertices of C are consecutive,*
- *K is a clique in $\mathcal{G}(\mathbf{X})$ and each $(i,j) \in K$ is adjacent to at least three vertices of C which are consecutive vertices of C,*

then $\mathcal{S}^Q(\mathbf{X})$ is a minimally non-firm binary matrix.

Proof. We show that the conditions of Theorem 5.3.1 hold for such $\mathbf{X}$ and Q.

As $\mathfrak{br}(\mathbf{X}) = k$, every matrix obtained from $\mathbf{X}$ by setting some 1s to ?s can be covered by at most k rectangles. On the other hand, for all $P \subseteq Q$ the rectangle cover graph of $\mathbf{X}^P$ contains the $2k-1$-hole C, so we have $\mathfrak{br}(\mathbf{X}^P) = k$ for all $P \subseteq Q$.

A maximum independent set of C shows that $\mathfrak{i}(\mathbf{X}^P) \geq k-1$ for all $P \subseteq Q$. For $P \subsetneq Q$, let $(i,j) \in Q \setminus P$. By requirement, (i,j) is adjacent to exactly two consecutive vertices of C. Let S be a maximum independent set of C that does not contain the two vertices that (i,j) is adjacent to. Then $S \cup \{(i,j)\}$ is a feasible isolated set of size k. Therefore, for all $P \subsetneq Q$ we have $\mathfrak{i}(\mathbf{X}^P) = k$.

For $\mathbf{X}^Q$ we have all the entries in Q set to ?s. Hence any isolated set of $\mathbf{X}^Q$ is a subset of $C \cup K$. Since K is a clique and each vertex in it is adjacent to at least three consecutive vertices of the $2k-1$-hole C, any isolated set $S \subset C \cup K$ satisfies $|S| \leq k-1$. Therefore, $\mathfrak{i}(\mathbf{X}^Q) = k-1$.

For all $P \subseteq Q$, if $\mathbf{Y}$ is a proper submatrix of $\mathbf{X}^P$, then $\mathbf{Y}$ is superfirm because $\mathbf{X}$ is mnsf.

Therefore, we have shown that $\mathbf{X}^Q$ is an mnf generalised binary matrix, and $\mathbf{X}^P$ is firm for all $P \subsetneq Q$, hence $\mathbf{X}$ and Q satisfy the conditions of Theorem 5.3.1 and $\mathcal{S}^Q(\mathbf{X})$ is an mnf binary matrix. □

Let us now use this theorem to show that we can obtain a minimally non-firm matrix from the firm mnsf matrices $\mathbf{D}_4, \mathbf{T}_5, \mathbf{M}_n$ $(n \geq 4)$ and $\mathbf{H}_n$ $(n \geq 3)$.

First we present two totally balanced square mnf matrices, one from $\mathbf{D}_4$ and one from $\mathbf{T}_5$. Let us partition $\text{supp}_1(\mathbf{D}_4)$ into three sets Q_{D_4}, K_{D_4}, C_{D_4} and $\text{supp}_1(\mathbf{T}_5)$ into Q_{T_5}, K_{T_5}, C_{T_5} given by

$$\begin{aligned} Q_{D_4} &= \{(1,1),(3,4),(4,3)\}, & Q_{T_5} &= \{(1,1),(4,5),(5,4)\},\\ K_{D_4} &= \{(2,2),(2,3),(3,2)\}, & K_{T_5} &= \{(2,2),(2,3),(3,2)\},\\ C_{D_4} &= \text{supp}_1(\mathbf{D}_4) \setminus (K_{D_4} \cup Q_{D_4}), & C_{T_5} &= \text{supp}_1(\mathbf{T}_5) \setminus (K_{T_5} \cup Q_{T_5}). \end{aligned}$$

To visualise these partitions, we indicate entries in C by red dots, entries in Q by ?s and entries in K by standard 1s,

$$\mathbf{D}_4^{Q_{D_4}} = \begin{bmatrix} ? & \bullet & & \\ \bullet & 1 & 1 & \bullet \\ & 1 & \bullet & ? \\ & \bullet & ? & \end{bmatrix}, \qquad \mathbf{T}_5^{Q_{T_5}} = \begin{bmatrix} ? & \bullet & & & \\ \bullet & 1 & 1 & & \bullet \\ & 1 & \bullet & \bullet & \\ & & \bullet & & ? \\ & \bullet & & ? & \end{bmatrix}.$$

One can then see that C_{D_4} is exactly the 5-hole that is shown in Figure 3.2 and C_{T_5} is the 7-hole that is highlighted in Figure 4.4. Furthermore, $\mathfrak{br}(\mathbf{D}_4) = 3$ by Lemma 4.4.4 and $\mathfrak{br}(\mathbf{T}_5) = 4$ by Lemma 4.4.5 and it is easy to verify that $K_{D_4}, Q_{D_4}, K_{T_5}, Q_{T_5}$ satisfy the conditions of Theorem 5.4.1. Therefore, we obtain the following result.

Theorem 5.4.2. *$\mathcal{S}^{Q_{D_4}}(\mathbf{D}_4) \in \{0,1\}^{7\times 7}$ and $\mathcal{S}^{Q_{T_5}}(\mathbf{T}_5) \in \{0,1\}^{8\times 8}$ are minimally non-firm.*

So how do these mnf matrices look like? Using the convention on the ordering of stretching (stretch entries in Q from left to right and then top to bottom), $\mathcal{S}^{Q_{D_4}}(\mathbf{D}_4)$ looks like,

$$\mathcal{S}^{Q_{D_4}}(\mathbf{D}_4) = \left[\begin{array}{cccc|ccc} 1 & 1 & & & 1 & & \\ 1 & 1 & 1 & 1 & & & \\ & 1 & 1 & 1 & & 1 & \\ & 1 & 1 & & & & 1 \\ \hline 1 & & & & 1 & & \\ & & & 1 & & 1 & \\ & & 1 & & & & 1 \end{array}\right],$$

and $\mathcal{S}^{Q_{T_5}}(\mathbf{T}_5)$ looks like

$$\mathcal{S}^{Q_{T_5}}(\mathbf{T}_5) = \left[\begin{array}{ccccc|ccc} 1 & 1 & & & & 1 & & \\ 1 & 1 & 1 & & 1 & & & \\ & 1 & 1 & 1 & & & & \\ & & 1 & & 1 & & 1 & \\ & 1 & & 1 & & & & 1 \\ \hline 1 & & & & & 1 & & \\ & & & & 1 & & 1 & \\ & & & 1 & & & & 1 \end{array}\right].$$

Observe that both mnf matrices, can be ordered to become symmetric by exchanging their last row with the penultimate one. $\mathcal{S}^{Q_{D_4}}(\mathbf{D}_4)$ is exactly the matrix that is presented in Equation (2.1.2) which is obtained from the swath matrix of Chung's polygon and is also shown in [77, Fig 1.1]. Furthermore, since stretching preserves totally balancedness by Lemma 4.3.11 and $\mathbf{D}_4$ is interval and $\mathbf{T}_5$ is totally balanced, both $\mathcal{S}^{Q_{D_4}}(\mathbf{D}_4)$ and $\mathcal{S}^{Q_{T_5}}(\mathbf{T}_5)$ are totally balanced.

Let us now present two infinite families of mnf matrices, one obtained from $\mathbf{M}_n$ ($n \geq 4$) and one from $\mathbf{H}_n$ ($n \geq 3$). Let us partition $\text{supp}_1(\mathbf{M}_n)$ into the three sets Q_{M_n}, K_{M_n}, C_{M_n} and $\text{supp}_1(\mathbf{H}_n)$ into Q_{H_n}, K_{H_n}, C_{H_n} given by

$$\begin{aligned} Q_{M_n} &= \{n, n\}, & Q_{H_n} &= \{(n,2),(n,n+1)\}, \\ K_{M_n} &= \{(n-1,n-1)\}, & K_{H_n} &= \{(\ell,1) : \ell \in [n-1]\}, \\ C_{M_n} &= \text{supp}_1(\mathbf{M}_n) \setminus (K_{M_n} \cup Q_{M_n}), & C_{H_n} &= \text{supp}_1(\mathbf{H}_n) \setminus (K_{H_n} \cup Q_{H_n}). \end{aligned}$$

We can again visualise these partitions by indicating entries in C by red dots, entries in Q by ?s and entries in K by standard 1s,

$$\mathbf{M}_n^{Q_{M_n}} = \begin{bmatrix} \bullet & \bullet & & & & \\ & \bullet & \bullet & & & \\ & & \ddots & \ddots & & \\ & & & \bullet & \bullet & \\ \bullet & & & & 1 & \bullet \\ & & & & \bullet & ? \end{bmatrix}, \qquad \mathbf{H}_n^{Q_{H_n}} = \begin{bmatrix} 1 & \bullet & \bullet & & & \\ 1 & & \bullet & \bullet & & \\ \vdots & & & \ddots & \ddots & \\ 1 & & & & \bullet & \bullet \\ \bullet & ? & & & & ? \end{bmatrix}.$$

Then C_{M_n} is exactly the $2n-1$-hole in $\mathcal{G}(\mathbf{M}_n)$ that is shown in Figure 3.7 for $n = 4, 5$ and C_{H_n} is the $2n-1$-hole in $\mathcal{G}(\mathbf{H}_n)$ that is highlighted in Figure 4.9 for $n = 4, 5$. Furthermore, $\mathfrak{br}(\mathbf{M}_n) = n$ by Lemma 4.4.2 and $\mathfrak{br}(\mathbf{H}_n) = n$ by Lemma 4.4.3 and it is easy to verify that their partitions satisfy the conditions of Theorem 5.4.1. Therefore, we obtain the following results.

Theorem 5.4.3. *For $n \geq 4$, $\mathcal{S}^{(n,n)}(\mathbf{M}_n) \in \{0,1\}^{(n+1)\times(n+1)}$ is minimally non-firm.*

Theorem 5.4.4. *For $n \geq 3$, $\mathcal{S}^{\{(n,2),(n,n+1)\}}(\mathbf{H}_n) \in \{0,1\}^{(n+2)\times(n+3)}$ is minimally non-firm.*

To the best of our knowledge, these mnf matrices have not appeared anywhere in the literature before and they are the first infinite family of matrices to be proved mnf. So how do they look like? $\mathcal{S}^{(n,n)}(\mathbf{M}_n)$ looks like

$$\mathcal{S}^{(n,n)}(\mathbf{M}_n) = \left[\begin{array}{cccccc|c} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & 1 & & \\ 1 & & & & 1 & 1 & \\ & & & & 1 & 1 & 1 \\ \hline & & & & & 1 & 1 \end{array}\right],$$

and $\mathcal{S}^{Q_{H_n}}(\mathbf{H}_n)$ has the form,

$$\mathcal{S}^{\{(n,2),(n,n+1)\}}(\mathbf{H}_n) = \left[\begin{array}{cccccc|cc} 1 & 1 & 1 & & & & & \\ 1 & & 1 & 1 & & & & \\ \vdots & & & \ddots & \ddots & & & \\ 1 & & & & 1 & 1 & & \\ 1 & 1 & & & & 1 & 1 & 1 \\ \hline & 1 & & & & & 1 & \\ & & & & & 1 & & 1 \end{array}\right].$$

It is possible to permute $\mathcal{S}^{(n,n)}(\mathbf{M}_n)$ into a symmetric form. On the other hand, $\mathcal{S}^{Q_{H_n}}(\mathbf{H}_n)$ is of dimension $(n+2)\times(n+3)$, the first non-square infinite family of mnf matrices. Clearly, the transpose of $\mathcal{S}^{Q_{H_n}}(\mathbf{H}_n)$ is also mnf.

Observe that Q_{D_4}, Q_{T_5} and Q_{M_n} consist of the simplicial 1s of $\mathbf{D}_4, \mathbf{T}_5$ and $\mathbf{M}_n$, respectively. If an mnsf matrix $\mathbf{X}$ has some simplicial 1, then each simplicial 1 must always be adjacent to exactly two consecutive vertices of the odd hole in $\mathcal{G}(\mathbf{X})$. On the other hand, none of the two 1s in Q_{H_n} are simplicial as $\mathbf{H}_n$ has no simplicial 1s.

There is one more mnsf family in the previous chapter from which we have not yet constructed an mnf family: $\mathbf{W}_n$ for $n \geq 5$. This mnsf family is slightly different from the others considered and no such partition of $\operatorname{supp}_1(\mathbf{W}_n)$ exists which would satisfy the conditions of Theorem 5.4.1. Furthermore, $\mathcal{G}(\mathbf{W}_n)$ contains a $2n-3$-hole but it has full Boolean rank and isolation number $\mathfrak{i}(\mathbf{W}_n) = \mathfrak{br}(\mathbf{W}_n) = n$ by Lemma 4.4.6, in contrast to all previous mnsf matrices which have rank equal to the clique cover number of their odd holes. The proof however, that we present to get an mnf family from $\mathbf{W}_n$ is very similar to that of Theorem 5.4.1. Let us partition $\operatorname{supp}_1(\mathbf{W}_n)$ into three sets Q_{W_n}, F_{W_n} and C_{W_n} given by

$$\begin{aligned} Q_{W_n} &= \{(n-2,n),(n-1,n-1),(n,n-2)\}, \\ F_{W_n} &= \{(n-3,n-2),(n-3,n-1),(n-2,n-3),(n-2,n-2),(n-1,n-3)\}, \\ C_{W_n} &= \operatorname{supp}_1(\mathbf{W}_n) \setminus (Q_{W_n} \cup F_{W_n}). \end{aligned}$$

We can visualise this partition similarly as before, indicating the entries in C_{W_n} by red dots, Q_{W_n} by ?s and entries in F_{W_n} by standard 1s,

$$\mathbf{W}_n^{Q_{W_n}} = \begin{array}{c|ccccccc|} & 1 & 2 & & n-3 & n-2 & n-1 & n \\ 1 & \bullet & \bullet & & & & & \\ 2 & \bullet & & \ddots & & & & \\ & & \ddots & & \bullet & & & \\ n-3 & & & \bullet & & 1 & 1 & \bullet \\ n-2 & & & & 1 & 1 & \bullet & ? \\ n-1 & & & & 1 & \bullet & ? & \\ n & & & & \bullet & ? & & \end{array}.$$

Then we can see that C_{W_n} contains the vertices of the $2n-3$-hole of $\mathbf{W}_n$ which is shown in Figure 4.10 for $n=5$. Furthermore, observe that every entry in Q_{W_n} is adjacent to exactly two consecutive vertices of C_{W_n}. In addition, every entry in F_{W_n} is also adjacent to at exactly four consecutive vertices of C_{W_n}, but F_{W_n} is not a clique.

Lemma 5.4.5. *For $n \geq 5$, $\mathcal{S}^{Q_{W_n}}(\mathbf{W}_n) \in \{0,1\}^{(n+3)\times(n+3)}$ is minimally non-firm.*

Proof. By Lemma 4.4.6 $\mathbf{W}_n$ is a firm mnsf matrix with $\mathfrak{i}(\mathbf{W}_n) = \mathfrak{br}(\mathbf{W}_n) = n$. In this proof, we show that $\mathfrak{i}(\mathbf{W}_n^{Q_{W_n}}) = n-2$ and $\mathfrak{br}(\mathbf{W}_n^{Q_{W_n}}) = \mathfrak{i}(\mathbf{W}_n^P) = \mathfrak{br}(\mathbf{W}_n^P) = n-1$ for all $\emptyset \neq P \subsetneq Q_{W_n}$. And then by Theorem 5.3.1 $\mathcal{S}^{Q_{W_n}}(\mathbf{W}_n)$ is minimally non-firm.

For all $P \subseteq Q_{W_n}$, $\mathcal{G}(\mathbf{W}_n^P)$ has the $2n-3$-hole C_{W_n}. Hence, for all $P \subseteq Q_{W_n}$ we have $\mathfrak{br}(\mathbf{W}_n^P) \geq n-1$. On the other hand, let us specify three maximal rectangles of $\mathbf{W}_n$,

$$\begin{aligned} R_1 &= \{n-2, n-1, n\} \times \{n-3, n-2\}, \\ R_2 &= \{n-3, n-2, n-1\} \times \{n-2, n-1\}, \\ R_3 &= \{n-3, n-2\} \times \{n-2, n-1, n\}. \end{aligned}$$

If we can cover $\mathbf{W}_n^P$ with $n-1$ rectangles for $P \subseteq Q_{W_n}$ with $|P| = 1$, then we can also cover $\mathbf{W}_n^P$ with $n-1$ rectangles for all $\emptyset \neq P \subseteq Q_{W_n}$. The bottom submatrix of $\mathbf{W}_n^P$ for the two non-symmetric cases of $P \subseteq Q_{W_n}$ with $|P| = 1$ is shown below,

$$\begin{array}{c|ccccc|} & & n{-}3 & n{-}2 & n{-}1 & n \\ & * & 1 & & & \\ n{-}3 & 1 & & 1 & 1 & 1 \\ n{-}2 & & 1 & 1 & 1 & ? \\ n{-}1 & & 1 & 1 & 1 & \\ n & & 1 & 1 & & \end{array}\;, \qquad \begin{array}{c|ccccc|} & & n{-}3 & n{-}2 & n{-}1 & n \\ & * & 1 & & & \\ n{-}3 & 1 & & 1 & 1 & 1 \\ n{-}2 & & 1 & 1 & 1 & 1 \\ n{-}1 & & 1 & 1 & ? & \\ n & & 1 & 1 & & \end{array}\;.$$

One can see that $\mathbf{W}_n^{(n-2,n)}$ can be covered by $n-3$ row rectangles for rows $1, \ldots, n-3$ plus rectangles R_1 and R_2; and $\mathbf{W}_n^{(n-1,n-1)}$ can be covered by $n-3$ row rectangles for rows $1, \ldots, n-3$ plus rectangles R_1 and R_3. Therefore, for all $\emptyset \neq P \subseteq Q_{W_n}$ we have $\mathfrak{br}(\mathbf{W}_n^P) = n-1$.

Each $(i,j) \in Q_{W_n}$ is adjacent to exactly two consecutive vertices of C_{W_n} and not adjacent to the other $2n-5$ vertices of the hole. For $P \subsetneq Q_{W_n}$, let $(\ell, k) \in Q_{W_n} \setminus P$. Then taking a maximum independent set of the $2n-3$ hole C_{W_n} which does not contain the two vertices that (ℓ, k) is adjacent to, and adding in (ℓ, k) gives an isolated set of size $n-1$. Hence, for all $\emptyset \neq P \subsetneq Q_{W_n}$ we have $\mathfrak{i}(\mathbf{W}_n^P) = n-1$.

Any maximum independent set of C_{W-n} gives an isolated set of size $n-2$ for $\mathbf{W}_n^{Q_{W_n}}$. Suppose that S_n is an isolated set of size $n-1$ for $\mathbf{W}_n^{Q_{W_n}}$. Then S_n contains a 1 from each row except for exactly one row, and a 1 from each column except for exactly one column. The bottom submatrix of $\mathbf{W}_n^{Q_{W_n}}$ is

$$\begin{array}{c} \\ \\ n{-}3 \\ n{-}2 \\ n{-}1 \\ n \end{array}\begin{array}{c} \begin{array}{ccccc} & n{-}3 & n{-}2 & n{-}1 & n \end{array} \\ \begin{bmatrix} * & 1 & & & \\ 1 & & 1 & 1 & 1 \\ & 1 & 1 & 1 & ? \\ & 1 & 1 & ? & \\ & 1 & ? & & \end{bmatrix} \end{array}.$$

Row n and column n both only have a single 1. (a) If $(n-3, n) \in S_n$, then column $n-1$ cannot have any 1s in S_n, so every column apart from $n-1$ must have an element in S_n. Similarly, from column $n-2$, S_n then can only contain $(n-1, n-2)$, so this must be in S_n. But then S_n cannot contain any 1s from row $n-2$ and n, so $|S_n| < n-1$. (b) If $(n, n-3) \in S_n$, the argument is symmetric hence $|S_n| < n-1$. So we have neither $(n-3, n)$ nor $(n, n-3)$ in S_n. Then S_n must have a 1 from every row and column with index in $[n-1]$. Columns $n-2$ and $n-1$ form a rectangle and also rows $n-2$ and $n-1$ form a rectangle, so S_n cannot contain a 1 from both and $|S_n| < n-1$. Hence $\mathfrak{i}(\mathbf{W}_n^{Q_{W_n}}) = n-2$.

If $\mathbf{Y}$ is a proper submatrix of $\mathbf{W}_n^P$ for any $P \subseteq Q_{W_n}$, then $\mathbf{Y}$ is superfirm because $\mathbf{W}_n$ is mnsf. Therefore all conditions of Theorem 5.3.1 are satisfied and $\mathcal{S}^{Q_{W_n}}(\mathbf{W}_n)$ is minimally non-firm. □

Let us show $\mathcal{S}^{Q_{W_n}}(\mathbf{W}_n)$ with our standard stretching ordering,

$$\mathcal{S}^{Q_{W_n}}(\mathbf{W}_n) = \left[\begin{array}{ccccccc|ccc} 1 & 1 & & & & & & & & \\ 1 & & \ddots & & & & & & & \\ & \ddots & & 1 & & & & & & \\ & & 1 & & 1 & 1 & 1 & & & \\ & & & 1 & 1 & 1 & 1 & 1 & & \\ & & & 1 & 1 & 1 & & & 1 & \\ & & & 1 & 1 & & & & & 1 \\ \hline & & & & & & 1 & 1 & & \\ & & & & & 1 & & & 1 & \\ & & & & 1 & & & & & 1 \end{array}\right].$$

If the last three rows are permuted, it can be brought to symmetric form.

We think that the proof of $\mathbf{W}_n$ could be generalised to mnsf matrices $\mathbf{X}$ similar in structure to $\mathbf{W}_n$ which have a partition of $\text{supp}_1(\mathbf{X})$ into three sets Q, F and C where C is an odd-hole of the mnsf matrix, Q has vertices adjacent to exactly two consecutive vertices of C, F contains vertices that are adjacent to at least four consecutive vertices of C and F can be covered by two cliques. Sadly, we do not know of any more mnsf matrices similar to $\mathbf{W}_n$ on which we could test this idea.

For all mnsf matrices in this section, Q is defined to be the subset $Q \subset \operatorname{supp}_1(\mathbf{X})$ which contains entries that are adjacent to exactly two consecutive vertices of an odd hole of mnsf $\mathbf{X}$. Furthermore, in all cases we had $\mathfrak{br}(\mathbf{X}^Q) = k$ and $\mathfrak{i}(\mathbf{X}^Q) = k - 1$ if the odd hole of $\mathbf{X}$ is of size $2k - 1$.

We suspect that these observations hold for any mnsf matrix and one can obtain an mnf matrix from every mnsf matrix by stretching the subset of 1s which contains entries that are adjacent to exactly two consecutive vertices of an odd hole of $\mathbf{X}$. Of course, for this suspicion to have a chance to be true, the conjecture that every mnsf matrix is firm should be proved first. Furthermore, it should also be shown that an mnsf matrix cannot have two odd holes of different sizes.

Conjecture 5.4.6. *If $\mathbf{X} \in \{0,1\}^{m\times n}$ is mnsf and has several odd holes, then every odd hole is of the same size. If $Q \subset \operatorname{supp}_1(\mathbf{X})$ contains every vertex that is adjacent to exactly two consecutive vertices of an odd hole in $\mathcal{G}(\mathbf{X})$, then $\mathcal{S}^Q(\mathbf{X})$ is minimally non-firm.*

5.5 Mnf matrices by 2-simplicial neighbour stretching

In this section, we present further infinite families of minimally non-firm matrices and show that the base matrix from which one can obtain a minimally non-firm matrix via Theorem 5.3.1 does not need to be minimally non-superfirm, but can contain several nested odd holes. Recall Definition 4.3.7 that (ℓ, k) is a 2-simplicial neighbour of an $m \times n$ matrix $\mathbf{X}$ if

(a) row ℓ contains exactly two nonzero entries (ℓ, k) and (ℓ, j) for some $j \in [n]$, and (ℓ, j) is a simplicial 1 of $\mathbf{X}$, or

(b) column k contains exactly two nonzero entries (ℓ, k) and (i, k) for some $i \in [m]$, and (i, k) is a simplicial 1 of $\mathbf{X}$.

Theorem 4.3.9 shows that 2-simplicial neighbour stretching preserves superfirmness, firmness and increases the size of certain odd holes. We will apply repeated 2-simplicial neighbour stretchings to mnsf matrices to get matrices which contain several nested odd holes.

Recall that whenever we stretch a 1 at (ℓ, k) of an $m \times n$ matrix $\mathbf{X}$, we create a simplicial 1 at $(m+1, n+1)$ which has a maximal rectangle of size 2×2 indexed by $\{\ell, m{+}1\} \times \{k, n{+}1\}$ and thus both $(\ell, n{+}1)$ and $(m{+}1, k)$ are 2-simplicial neighbours in $\mathcal{S}^{(\ell,k)}(\mathbf{X})$. Thus 2-simplicial neighbour stretching can be applied repeatedly. Let $\mathcal{S}^{(\ell_2,k_2)} \circ \mathcal{S}^{(\ell_1,k_1)}(\mathbf{X})$ be a shorthand notation for $\mathcal{S}^{(\ell_2,k_2)}(\mathcal{S}^{(\ell_1,k_1)}(\mathbf{X}))$. We say that

$$\mathcal{S}^{(\ell_s,k_s)} \circ \mathcal{S}^{(\ell_{s-1},k_{s-1})} \circ \ldots \circ \mathcal{S}^{(\ell_1,k_1)}(\mathbf{X})$$

is a *chain of* 2*-simplicial neighbour stretchings* if (ℓ_1, k_1) is 2-simplicial neighbour in $\mathbf{X}$ and for all $i \geq 2$

$$\text{either} \quad \ell_i = \ell_{i-1},\ k_i \neq k_{i-1}, \qquad \text{or} \quad \ell_i \neq \ell_{i-1},\ k_i = k_{i-1}.$$

For instance, $\mathcal{S}^{(\ell_7,k_7)} \circ \ldots \circ \mathcal{S}^{(\ell_2,k_2)} \circ \mathcal{S}^{(\ell_1,k_1)}(\mathbf{J}_2)$ is a chain of 2-simplicial neighbour stretchings,

$$\begin{array}{c|ccccccccc|}
 & & k_1,k_2 & & k_3 & k_4 & k_5,k_6 & & k_7 & \\
\ell_1 & 1 & 1 & 1 & & & & & & \\
 & 1 & 1 & & & & & & & \\
\ell_2,\ell_3,\ell_4,\ell_5 & & 1 & 1 & 1 & 1 & 1 & 1 & & \\
 & & 1 & & 1 & & & & & \\
 & & & & 1 & 1 & & & & \\
 & & & & & 1 & 1 & & & \\
\ell_6,\ell_7 & & & & & & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & & 1 & \\
 & & & & & & & & 1 & 1
\end{array}. \tag{5.5.1}$$

Observe that the chain *'can change direction' at the i-th stretching* if

$$\begin{array}{rl}
\text{if} & \ell_{i-1} = \ell_{i-2}, k_{i-1} \neq k_{i-2} \quad \text{and} \quad \ell_i \neq \ell_{i-1}, k_i = k_{i-1}, \\
\text{or} & \ell_{i-1} \neq \ell_{i-2}, k_{i-1} = k_{i-2} \quad \text{and} \quad \ell_i = \ell_{i-1}, k_i \neq k_{i-1}.
\end{array}$$

For instance, in Equation (5.5.1) the direction is changed at the 3rd, 6th and 7th stretchings.

Chains of 2-simplicial neighbour stretchings are useful because they can be used to obtain matrices with arbitrarily large odd holes. For instance, applying three 2-simplicial neighbour stretchings to mnsf matrix $\mathbf{M}_4$ we get the matrix

$$\mathcal{S}^{(5,6)} \circ \mathcal{S}^{(5,4)} \circ \mathcal{S}^{(3,4)}(\mathbf{M}_4) = \begin{bmatrix}
1 & 1 & & & & & \\
 & 1 & 1 & & & & \\
1 & & 1 & 1 & 1 & & \\
 & & 1 & 1 & & & \\
 & & & 1 & 1 & 1 & 1 \\
 & & & 1 & & 1 & \\
 & & & & & 1 & 1
\end{bmatrix},$$

which has a $7+2\cdot 3 = 13$-hole that is obtained by adding two vertices to the 7-hole of $\mathbf{M}_4$ at each stretching. We show the rectangle cover graph of this matrix in Figure 5.1 and highlight its 13-hole. Note that this matrix contains several nested odd-holes, the 7-hole of $\mathbf{M}_4$, a 9-hole and an 11-hole too. Also observe that as shown by Theorem 4.3.9, $\mathcal{S}^{(5,6)} \circ \mathcal{S}^{(5,4)} \circ \mathcal{S}^{(3,4)}(\mathbf{M}_4)$ is firm and has exactly one simplicial 1 at $(7,7)$.

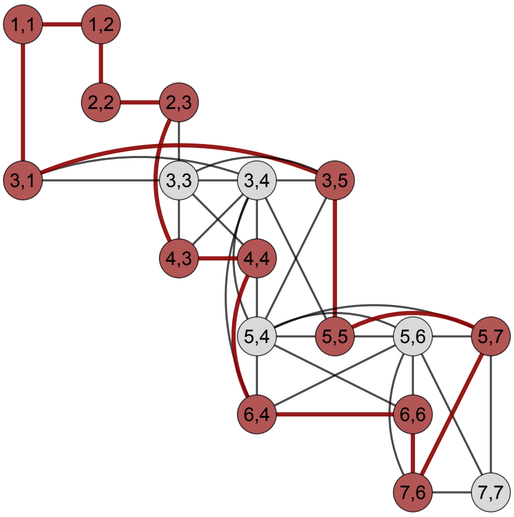


Figure 5.1: The 13-hole in the rectangle cover graph of $\mathcal{S}^{(5,6)} \circ \mathcal{S}^{(5,4)} \circ \mathcal{S}^{(3,4)}(\mathbf{M}_4)$

In the rest of this section we show that stretching the simplicial 1s of a matrix obtained from any of the mnsf matrices $\mathbf{M}_n$, $\mathbf{D}_4$ and $\mathbf{T}_5$ via a series of 2-simplicial neighbour stretchings gives a minimally non-firm binary matrix. For this we will need to argue that proper submatrices of such matrices are firm. In the previous section this step was easily done as the base matrix was mnsf and that implied any proper submatrix to be superfirm. Here in this section however, to prove that proper submatrices are firm requires more work as the base matrix contains several nested odd-holes.

Recall from Section 3.3.2, that a matrix $\mathbf{X}$ can be L-decomposed if its rows and columns can be partitioned into the block form,

$$\mathbf{X} = \begin{array}{c c} & \begin{array}{ccc} J_A^0 & J_A^1 & J_B \end{array} \\ \begin{array}{c} I_A \\ I_B^1 \\ I_B^0 \end{array} & \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \\ & \mathbf{J} & \mathbf{B}_1 \\ & & \mathbf{B}_0 \end{bmatrix} \end{array}.$$

Then $\mathbf{X}$ is the L-sum of two submatrices $\mathbf{A}$, $\mathbf{B}$ of $\mathbf{X}$,

$$\mathbf{X} = \mathcal{L}^{(i_A,j_B)}(\mathbf{A},\mathbf{B}) = \begin{array}{c c} & \begin{array}{cc} J_A^0 & J_A^1 \end{array} \\ \begin{array}{c} I_A \\ i_A \end{array} & \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \\ \mathbf{0}^\top & \mathbf{1}^\top \end{bmatrix} \end{array} \odot \begin{array}{c c} & \begin{array}{cc} j_B & J_B \end{array} \\ \begin{array}{c} I_B^1 \\ I_B^0 \end{array} & \begin{bmatrix} \mathbf{1} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{B}_0 \end{bmatrix} \end{array}.$$

In the next lemma, we show that some specific proper submatrices of a matrix that is obtained by a chain of 2-simplicial neighbour stretchings can always be written as the L-sum or direct sum of two smaller matrices. This lemma will be used in the theorems that follow, to show that such proper submatrices then cannot be mnf. We only consider square matrices here for simplicity, because stretching a square matrix results in a square matrix, which has a simplicial 1 at the bottom diagonal entry due to our stretching ordering convention.

Lemma 5.5.1. *Let $\mathbf{X} \in \{0,1\}^{n\times n}$ be obtained from an $m \times m$ binary matrix via a chain of $n - m > 1$ 2-simplicial neighbour stretchings. Then $(n,n) \in \operatorname{supp}_1(\mathbf{X})$ is a simplicial 1 created by the last stretching. If $\mathbf{Y}$ is a proper submatrix of $\mathbf{X}$ or $\mathbf{X}^{(n,n)}$ indexed by $I \times J$ such that*

- *$[m] \subset I$, $[m] \subset J$ and $n \in I$, $n \in J$, and*
- *$\mathbf{Y}$ has at least two non-zeroes in each row and column,*

then $\mathbf{Y}$ is the direct sum or L-sum of two of its proper submatrices, one containing submatrix $[m] \times [m]$ and the other containing entry (n,n).

Proof. Since $\mathbf{Y}$ is a proper submatrix of $\mathbf{X}$ or $\mathbf{X}^{(n,n)}$ we must have $(p,p) \notin I \times J$ for some $m < p < n$. Observe that for each $m < p < n$, exactly one of row p or column p has exactly two 1s. If row p has exactly two 1s, then $(p,p) \notin I \times J$ implies $p \notin I$, else if column p has exactly two 1s, then $p \notin J$. Let (ℓ,k) be the 1 whose stretching created (p,p).

(a) Let row p have exactly two 1s, so $p \notin I$. If $\ell \notin I$ then $\mathbf{Y}$ is the direct sum of two of its non-empty proper submatrices indexed by $I_1 \times J_1$ and $I_2 \times J_2$ with

$$\begin{aligned} I_1 &= \{i \in I : i < p\}, & J_1 &= \{j \in J : j < p\}, \\ I_2 &= \{i \in I : i > p\}, & J_2 &= \{j \in J : j \geq p\}. \end{aligned}$$

On the other hand, if $\ell \in I$, let $T = \{t : (\ell,t) \in \operatorname{supp}_1(\mathbf{Y}), p < t\}$. Then $I \times J$ can be partitioned as,

$$\begin{aligned} I_B^0 &= \{i \in I : i < p\} \setminus \{\ell\}, & J_B &= \{j \in J : j < p\}, \\ I_B^1 &= \{\ell\}, & J_A^1 &= \{p\} \cup T, \\ I_A &= \{i \in I : i > p\}, & J_A^0 &= \{j \in J : j > p\} \setminus T. \end{aligned}$$

For instance, the matrix below is a possible submatrix of $\mathbf{X}$ and if $p \notin I$, it can be L-decomposed on the rectangle that is shaded by grey,

$$\begin{array}{c|cccc|ccc|cc|}
 & & & & k & p & T & T & & \\
 & 1 & 1 & 1 & & & & & & \\
 & 1 & 1 & & & & & & & \\
\hline
\ell & & 1 & 1 & 1 & \colorbox{lightgray}{1} & \colorbox{lightgray}{1} & \colorbox{lightgray}{1} & & \\
\hline
 & & 1 & & 1 & & & & & \\
p & & & & \cancel{1} & \cancel{1} & & & & \\
\hline
 & & & & & 1 & 1 & & & \\
 & & & & & & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & & 1 & \\
 & & & & & & & & 1 & 1
\end{array}.$$

The partitioning satisfies the conditions of L-decomposition, hence $\mathbf{Y}$ is the L-sum of two of its proper submatrices indexed by

$$(I_A \cup \{\ell\}) \times (J_A^0 \cup J_A^1) \quad \text{and} \quad (\{\ell\} \cup I_B^0) \times (\{p\} \cup J_B).$$

Case (b) is symmetric. Let column p have only two 1s. If $k \notin J$, then $\mathbf{Y}$ is the direct sum of two of its proper submatrices indexed by $I_1 \times J_1$ and $I_2 \times J_2$ with

$$\begin{aligned}
I_1 &= \{i \in I : i < p\}, & J_1 &= \{j \in J : j < p\}, \\
I_2 &= \{i \in I : i \geq p\}, & J_2 &= \{j \in J : j > p\}.
\end{aligned}$$

If $k \in J$ let $T = \{t : (t, k) \in \operatorname{supp}_1(\mathbf{Y}), p < t\}$. But then the below partitioning L-decomposes $\mathbf{Y}$,

$$\begin{aligned}
I_A &= \{i \in I : i < p\}, & J_A^0 &= \{j \in J : j < p\} \setminus \{k\}, \\
I_B^1 &= \{p\} \cup T, & J_A^1 &= \{k\}, \\
I_B^0 &= \{i \in I : i > p\} \setminus T, & J_B &= \{j \in J : j > p\}.
\end{aligned}$$

For instance, the L-decomposition may look like this,

$$\begin{array}{c|cc|c|ccc|ccc|}
 & & & k & & & p & & & \\
 & 1 & 1 & 1 & 1 & & & & & \\
 & 1 & 1 & & & & & & & \\
 & & 1 & 1 & & & & & & \\
 & & & 1 & 1 & 1 & & & & \\
\ell & & & 1 & & 1 & \cancel{1} & & & \\
\hline
p & & & \colorbox{lightgray}{1} & & & \cancel{1} & 1 & & \\
T & & & \colorbox{lightgray}{1} & & & & 1 & 1 & 1 \\
T & & & \colorbox{lightgray}{1} & & & & & 1 & \\
\hline
 & & & & & & & & 1 & 1
\end{array}.$$

Hence $\mathbf{Y}$ is the L-sum of two of its proper submatrices indexed by

$$(I_A \cup \{p\}) \times (J_A^0 \cup \{k\}) \quad \text{and} \quad (I_B^1 \cup I_B^0) \times (\{k\} \cup J_B).$$

Therefore, in all cases $\mathbf{Y}$ is the direct- or L-sum of two of its proper submatrices. □

5.5.1 Further infinite families from $\mathbf{M}_n$

We are ready to show that several further infinite families of mnf matrices can be obtained from the mnsf matrices $\mathbf{M}_n$.

Theorem 5.5.2. *Let $m \geq 4$ and $n > m$. If $\mathbf{X} \in \{0,1\}^{n \times n}$ is obtained by a series of 2-simplicial neighbour stretchings from $\mathbf{M}_m$, then $\mathcal{S}^{(n,n)}(\mathbf{X})$ is a minimally non-firm binary matrix.*

Proof. $\mathbf{M}_n$ is firm and 2-simplicial neighbour stretchings preserve firmness by Theorem 4.3.9 so $\mathbf{X}$ is a firm matrix. We will show that $\mathbf{X}^{(n,n)}$ is a minimally non-firm generalised binary matrix and then by Theorem 5.3.1 $\mathcal{S}^{(n,n)}(\mathbf{X})$ is mnf.

First, note that since $\mathbf{M}_m$ satisfies the conditions of Part (4.) of Theorem 4.3.9, 2-simplicial neighbour stretching preserves the number of simplicial 1s of $\mathbf{M}_m$ and thus $\mathbf{X}$ has exactly one simplicial 1 at (n,n).

$\mathbf{X}^{(n,n)}$ has dimension $n \times n$ and $\mathcal{G}(\mathbf{X}^{(n,n)})$ has a $2n-1$ hole by Part (2.) of Theorem 4.3.9, so $\mathfrak{br}(\mathbf{X}^{(n,n)}) = n$.

A maximum independent set of the $2n-1$ hole gives $\mathfrak{i}(\mathbf{X}^{(n,n)}) \geq n-1$. Suppose S is an isolated set of $\mathbf{X}^{(n,n)}$ of size n. Then S contains exactly one 1 from each row and column of $\mathbf{X}^{(n,n)}$. Let (ℓ,k) be the 1 whose stretching created the 1 at (n,n) of $\mathbf{X}$. In $\mathbf{X}^{(n,n)}$, row n only has a single 1 at (n,k), so $(n,k) \in S$, while column n, also only has a single 1 at (ℓ,n), so $(\ell,n) \in S$. But this shows that S is not an isolated set as (n,k) and (ℓ,n) are in a common rectangle indexed by $\{\ell,n\} \times \{k,n\}$. Therefore, $\mathfrak{i}(\mathbf{X}^{(n,n)}) = n-1$.

Suppose that $\mathbf{X}^{(n,n)}$ has a proper submatrix that is not firm. Let $\mathbf{Y}$ be the smallest such submatrix of $\mathbf{X}^{(n,n)}$ indexed by $I \times J$, so $\mathbf{Y}$ is minimally non-firm. Since 2-simplicial neighbour stretching preserves firmness, $\mathbf{Y}$ cannot be a submatrix of $\mathbf{X}$, so we must have $n \in I$ and $n \in J$.

Recall that $\mathbf{M}_m$ is mnsf, so every proper submatrix of it is superfirm. Since, 2-simplicial neighbour stretching preserves superfirmness, if any of the first m rows or columns is missing in $\mathbf{Y}$, then $\mathbf{Y}$ is obtained by setting a 1 to a ? of a proper submatrix of the superfirm matrix that is obtained by repeated 2-simplicial neighbour stretchings of a proper submatrix of $\mathbf{M}_n$. So we must have $[m] \subseteq I$ and $[m] \subseteq J$.

Therefore we have $[m] \cup \{n\} \subset I$ and $[m] \cup \{n\} \subset J$. If $n = m + 1$, then $\mathbf{Y}$ is not a proper submatrix of $\mathbf{X}^{(n,n)}$, so we must have $n > m + 1$. But then by Lemma 5.5.1 $\mathbf{Y}$ is the direct or L-sum of two smaller submatrices of $\mathbf{X}^{(n,n)}$ indexed by $I_1 \times J_1$ and $I_2 \times J_2$ with $[m] \times [m] \subset I_1 \times J_1$ and $(n, n) \in I_2 \times J_2$. Since both direct sum and L-sum preserve firmness, no mnf matrix can be written as the direct or L-sum of any of its proper submatrices. Therefore, in any case $\mathbf{Y}$ is not mnf but firm and thus $\mathbf{X}^{(n,n)}$ is an mnf generalised binary matrix and $\mathcal{S}^{(n,n)}(\mathbf{X})$ is an mnf standard binary matrix. □

So how many new mnf matrices can we get by the above theorem? $\mathbf{M}_m$ has two 2-simplicial neighbours at $(m-1, m)$ and $(m, m-1)$, but $\mathbf{M}_m$ can be made symmetric, so $\mathcal{S}^{(m-1,m)}(\mathbf{M}_m)$ and $\mathcal{S}^{(m,m-1)}(\mathbf{M}_m)$ are permutations of each other. $\mathcal{S}^{(m-1,m)}(\mathbf{M}_m)$ however, is not symmetric and also has two 2-simplicial neighbours. So we can get two different matrices by 2-simplicial neighbour stretching from $\mathcal{S}^{(m-1,m)}(\mathbf{M}_m)$. This remains true for any further stretchings, and at each new stretching we can choose between two 1s to stretch. For instance, the number of mnf matrices that can be obtained by Theorem 5.4.3 and 5.5.2 from mnsf matrices $\mathbf{M}_m$ of dimension

- 5×5, is only one $\mathcal{S}^{(4,4)}(\mathbf{M}_4)$,
- 6×6, is $1 + 1$: $\mathcal{S}^{(5,5)}(\mathbf{M}_5)$ and $\mathcal{S}^{(5,5)} \circ \mathcal{S}^{(3,4)}(\mathbf{M}_4)$,
- 7×7, is $1 + 1 + 2$: $\mathcal{S}^{(6,6)}(\mathbf{M}_6)$, $\mathcal{S}^{(6,6)} \circ \mathcal{S}^{(5,6)}(\mathbf{M}_5)$, $\mathcal{S}^{(6,6)} \circ \mathcal{S}^{(3,5)} \circ \mathcal{S}^{(3,4)}(\mathbf{M}_4)$ and $\mathcal{S}^{(6,6)} \circ \mathcal{S}^{(5,4)} \circ \mathcal{S}^{(3,4)}(\mathbf{M}_4)$.

In general, for $n \geq 6$ the number of $n \times n$ mnf matrices that can be obtained from mnsf matrices $\mathbf{M}_m$ is $1 + \sum_{k=0}^{n-6} 2^k$.

5.5.2 Infinite families of mnf matrices from $\mathbf{D}_4$ and $\mathbf{T}_5$

Let us now apply 2-simplicial neighbour stretchings to create more mnf matrices from $\mathbf{D}_4$,

$$\mathbf{D}_4 = \begin{bmatrix} 1 & 1 & & \\ 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 \\ & 1 & 1 & \end{bmatrix}.$$

$\mathbf{D}_4$ has three simplicial 1s at $(1, 1), (3, 4), (4, 3)$ and it has four 2-simplicial neighbours at $(1, 2), (2, 1), (2, 4), (4, 2)$. Hence we may apply repeated 2-simplicial neighbour stretching to $\mathbf{D}_4$ to get matrices that have some nested odd holes. Observe that any matrix that is obtained by a series of 2-simplicial neighbour stretching from $\mathbf{D}_4$ has one, two or three *chains* of 2-simplicial neighbour stretchings which all come from different 'angles' and only the core matrix $\mathbf{D}_4$ links them together. For instance, the rectangle cover graph of

$$\mathcal{S}^{(9,2)} \circ \mathcal{S}^{(4,2)}(\mathcal{S}^{(7,4)} \circ \mathcal{S}^{(2,4)}(\mathcal{S}^{(1,5)} \circ \mathcal{S}^{(1,2)}(\mathbf{D}_4)))$$

consists of three chains of two 2-simplicial neighbour stretchings, one originating from $(1,2)$, one from $(2,4)$ and the third from $(4,2)$. In Figure 5.2 we show the rectangle cover graph of this matrix with rows and columns rearranged in a way to emphasise that each chain is independent from the other. We also highlight the $5+2\cdot 6 = 17$-hole in the graph which is the result of extending the 5-hole of $\mathbf{D}_4$ by 2 vertices at each stretching.

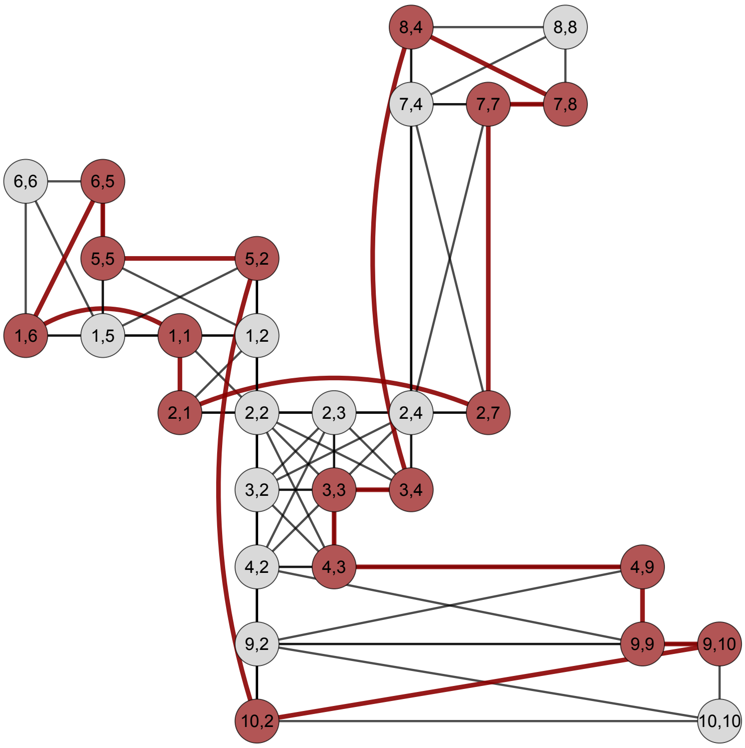


Figure 5.2: The rearranged rectangle cover graph of a matrix obtained from $\mathbf{D}_4$ by six 2-simplicial neighbour stretchings, which can be divided into three chains, each consisting of two stretchings

In the next theorem we show that stretching the three simplicial 1s of any matrix obtained from $\mathbf{D}_4$ by 2-simplicial neighbour stretchings gives an mnf matrix.

Theorem 5.5.3. *Let* $\mathbf{X} \in \{0,1\}^{n\times n}$ *be obtained by a series of* 2*-simplicial neighbour stretchings from* $\mathbf{D}_4$ *and let* Q *contain the indices of simplicial* 1*s of* $\mathbf{X}$*. Then* $\mathcal{S}^Q(\mathbf{X})$ *is a minimally non-firm binary matrix.*

Proof. We will show that $\mathbf{X}^P$ is firm for all $P \subsetneq Q$ and $\mathbf{X}^Q$ is mnf. Then Theorem 5.3.1 will imply that $\mathcal{S}^Q(\mathbf{X})$ is mnf.

Recall from Theorem 4.3.9 that simplicial neighbour stretching preserves firmness and the number of simplicial 1s if applied to a matrix without repeated rows and columns and with at least two 1s in each row and column. Since $\mathbf{D}_4$ is firm and satisfies these conditions, $\mathbf{X}$ is firm and has exactly three simplicial 1s, so $|Q| = 3$.

I. Since $\mathfrak{br}(\mathbf{D}_4) = 3$ and each 2-simplicial neighbour stretching increases the Boolean rank, the number of rows and the number of columns by exactly one, $\mathfrak{br}(\mathbf{X}^P) \leq \mathfrak{br}(\mathbf{X}) = 3 + n - 4 = n - 1$ for all $P \subseteq Q$. On the other hand, the 5-hole satisfies the conditions of part (2.) of Theorem 4.3.9 so $\mathcal{G}(\mathbf{X}^P)$ contains a $2n - 3$-hole for all $P \subseteq Q$, and $\mathfrak{br}(\mathbf{X}^P) = n - 1$.

II. Observe that all three simplicial 1s in Q are adjacent to exactly two vertices of the $2n - 3$-hole of $\mathcal{G}(\mathbf{X})$. For all $P \subsetneq Q$, let $(i, j) \in Q \setminus P$. Then taking a maximum independent set of the $2n - 3$-hole which does not contain the two vertices that (i, j) is adjacent to, and adding in (i, j) gives an isolated set of size $n - 1$ for all $\mathbf{X}^P$, $P \subsetneq Q$.

III. For $\mathbf{X}^Q$, all entries in Q are ?'s so we only have $\mathfrak{i}(\mathbf{X}^Q) \geq n - 2$ by any maximum independent set of the $2n - 3$-hole. Suppose $\mathbf{X}^Q$ has an isolated set S of size $n - 1$. Then S does not contain a 1 from exactly one row and exactly one column of $\mathbf{X}^Q$.

Recall that the simplicial 1 at $(1, 1)$ of $\mathbf{D}_4$ has a 2×2 maximal rectangle, the one at $(3, 4)$ has a 2×3 maximal rectangle and the one at $(4, 3)$ has a 3×2 maximal rectangle. By stretching we always create a simplicial 1 that has a 2×2 maximal rectangle and is on the diagonal of $\mathbf{X}$. Therefore, Q contains three simplicial 1s of $\mathbf{X}$,

- one of which has a 2×2 maximal rectangle and is on the diagonal, let this be (i_1, i_1) and its maximal rectangle $\{\ell_1, i_1\} \times \{k_1, i_1\}$;
- another one which has a maximal rectangle with exactly two columns, let this be (i_2, j_2) and its rectangle be $(I_2 \cup \{i_2\}) \times \{k_2, j_2\}$ with $|I_2| \in \{1, 2\}$;
- and another one which has a maximal rectangle with exactly two rows, let this be (i_3, j_3) and its rectangle be $\{\ell_3, i_3\} \times (J_3 \cup \{j_3\})$ with $|J_3| \in \{1, 2\}$.

Then in $\mathbf{X}^Q$, rows i_1 and i_2 have exactly one 1 at (i_1, k_1) and (i_2, k_2), respectively; and columns i_1 and j_3 also have exactly one 1 at (ℓ_1, i_1) and (ℓ_3, j_3), respectively.

If $S \cap \{(i_1, k_1), (\ell_1, i_1)\} = \emptyset$, then S is an isolated set of the $(n-1) \times (n-1)$ matrix $\mathbf{X}'^{P'}$ obtained by dropping row i_1 and column i_1 from $\mathbf{X}^Q$ and with $P' = Q \setminus \{(i_1, i_1)\}$.

But then $\mathbf{X}'$ is obtained by $n-5$ 2-simplicial neighbour stretchings from $\mathbf{D}_4$ and $\mathfrak{i}(\mathbf{X}'^{P'}) = n-2$ by Part **II.** Therefore, we must have $|S \cap \{(i_1, k_1), (\ell_1, i_1)\}| = 1$.

Since the cases $(i_1, k_1) \in S$ and $(\ell_1, i_1) \in S$ are symmetric, let $(\ell_1, i_1) \in S$. Next we explain why then S must contain the entries indicated by red dots,

$$\begin{array}{c|ccccccc|}
 & k_1 & i_1 & k_2 & j_2 & & & j_3 \\
\ell_1 & 1 & \bullet & & & & & \\
i_1 & \xcancel{1} & ? & & & & & \\
 & & & 1 & \xcancel{1} & & & \\
 & & & 1 & \xcancel{1} & & & \\
i_2 & & & \bullet & ? & & & \\
\ell_3 & & & & & 1 & 1 & \bullet \\
i_3 & & & & & \xcancel{1} & \xcancel{1} & ?
\end{array}.$$

If $(\ell_1, i_1) \in S$, then $(i_1, k_1) \notin S$, and S does not contain any 1s from row i_1, so it must contain a 1 from every other row of $\mathbf{X}^Q$. In particular, S must contain (i_2, k_2), the only 1 in row i_2. Then however, S cannot contain any 1s from column j_2, because all the 1s there are in a common rectangle with (i_2, k_2) (as (i_2, j_2) is a simplicial 1 in $\mathbf{X}$). So S must contain a 1 from every other column of $\mathbf{X}^Q$, in particular it must contain (ℓ_3, j_3), the only 1 in column j_3. If $\ell_1 = \ell_3$ then S cannot be an isolated set, so assume that $\ell_1 \neq \ell_3$. However, at this point S can also not contain any 1s from row i_3, as all those 1s are in a common rectangle with (ℓ_3, j_3). Hence, S is not an isolated set as it contains $n-1$ entries from $n-2$ rows. Therefore, no isolated set of size $n-1$ can exist for $\mathbf{X}^Q$ and $\mathfrak{br}(\mathbf{X}^Q) = n-2$.

IV. Let n be the smallest dimension for which $\mathbf{X}^P \in \{0, 1, ?\}^{n \times n}$ for some $P \subseteq Q$ has a non-firm proper submatrix. Let $\mathbf{Y}$ be a smallest non-firm proper submatrix of $\mathbf{X}^P$ indexed by $I \times J$. Then $\mathbf{Y}$ is mnf.

If $[4] \times [4] \not\subset I \times J$, then $\mathbf{Y}$ is equal to a

- a standard binary matrix $\mathbf{Z}$ that is obtained by a series of 2-simplicial neighbour stretchings applied to a proper submatrix of $\mathbf{D}_4$,
- or a proper submatrix of $\mathbf{Z}$,
- or a generalised binary matrix obtained from $\mathbf{Z}$ or one of its proper submatrices.

Since $\mathbf{D}_4$ is mnsf, any proper submatrix of it is superfirm. Furthermore, 2-simplicial neighbour stretching preserves superfirmness so any of the above options leads to a superfirm matrix. Therefore, we must have $[4] \times [4] \subset I \times J$.

If $Q \not\subset I \times J$ then $\mathbf{Y}$ is either equal to

- a matrix $\mathbf{X}'^{P'} \in \{0,1,?\}^{k\times k}$ for some $k < n$, where $\mathbf{X}'$ is obtained by $k-4$ 2-simplicial neighbour stretchings from $\mathbf{D}_4$ and $P' \subsetneq Q$, where P' is a subset of simplicial 1s of $\mathbf{X}'$;
- or a proper submatrix of $\mathbf{X}'^{P'}$.

In the first case, $\mathbf{X}'^{P'}$ with $P' \subsetneq Q$ has $\mathfrak{i}(\mathbf{X}'^{P'}) = \mathfrak{br}(\mathbf{X}'^{P'})$ by part **II.**, while in the second case the proper submatrix is firm by the minimality assumption on n. Therefore, we must have $Q \subset I \times J$ and $[4]\times[4] \subset I \times J$. Note that $\mathbf{X}$ is built up using possibly three different chains of 2-simplicial neighbour stretchings. In $\mathbf{Y}$ at least one of those chains must have a missing row or column, and thus by Lemma 5.5.1 $\mathbf{Y}$ can be decomposed into the direct- or L-sum of two of its proper submatrices. Since no mnf matrix can have such decomposition, $\mathbf{Y}$ must be firm and thus $\mathbf{X}^P$ is firm for all $P \subsetneq Q$ and $\mathbf{X}^Q$ is mnf. □

This theorem shows that we can obtain an infinite number of minimally non-firm matrices which all contain the mnsf matrix $\mathbf{D}_4$. In addition, since stretching preserves totally balancedness and $\mathbf{D}_4$ is an interval matrix, all these mnf matrices are totally balanced. One of the matrices that Theorem 5.5.3 proves to be mnf is the second one of the two matrices that Lubiw mentions in [77, Fig 1.1] to be not firm,

$$\mathcal{S}^{\{(1,1),(4,3),(5,5)\}}\mathcal{S}^{(2,4)}(\mathbf{D}_4) = \left[\begin{array}{ccccc|ccc} 1 & 1 & & & & 1 & & \\ 1 & 1 & 1 & 1 & 1 & & & \\ & 1 & 1 & 1 & & & & \\ & 1 & 1 & & & & 1 & \\ & & & 1 & 1 & & & 1 \\ \hline 1 & & & & & 1 & & \\ & & 1 & & & & 1 & \\ & & & & 1 & & & 1 \end{array}\right].$$

This example of Lubiw was also an inspiration to us to define stretching and to use it to extend the size of odd-holes of mnsf matrices.

So how many matrices exactly does Theorem 5.5.3 prove to be mnf? To understand this, for each dimension let us count how many matrices can be obtained from $\mathbf{D}_4$ by a series of 2-simplicial neighbour stretchings. Then all these matrices have three simplicial 1s which need to be stretched to get an mnf matrix, which increases the dimension by 3.

Since $\mathbf{D}_4$ is symmetric, with one 2-simplicial neighbour stretching we can get 2 different 5×5 matrices which are $\mathcal{S}^{(1,2)}(\mathbf{D}_4)$ and $\mathcal{S}^{(2,4)}(\mathbf{D}_4)$. With two 2-simplicial neighbour stretchings from $\mathbf{D}_4$, we can get 4 matrices with one chain of length 2,

$$\begin{aligned} &\mathcal{S}^{(1,5)} \circ \mathcal{S}^{(1,2)}(\mathbf{D}_4), &\qquad& \mathcal{S}^{(5,2)} \circ \mathcal{S}^{(1,2)}(\mathbf{D}_4), \\ &\mathcal{S}^{(2,5)} \circ \mathcal{S}^{(2,4)}(\mathbf{D}_4), && \mathcal{S}^{(5,4)} \circ \mathcal{S}^{(2,4)}(\mathbf{D}_4); \end{aligned}$$

and 3 matrices with two chains of length 1,

$$\mathcal{S}^{\{(1,2),(2,4)\}}(\mathbf{D}_4), \qquad \mathcal{S}^{\{(1,2),(4,2)\}}(\mathbf{D}_4), \qquad \mathcal{S}^{\{(2,4),(4,2)\}}(\mathbf{D}_4)^\top,$$

the last of which can be arranged to be a symmetric matrix. Hence there are 7 different 6×6 matrices obtained from $\mathbf{D}_4$ by 2-simplicial neighbour stretchings.

Applying three 2-simplicial neighbour stretchings can have one single chain of length 3 which we denote by $1 \circ 1 \circ 1$, or two chains: one of length one and the other of length two denoted by $1 \circ 2$, or three chains of length one. This leads to a total of $8 + (4 + 4 + 2) + 1 = 19$ different 7×7 matrices from $\mathbf{D}_4$. This counting takes into consideration that there are

- 2 choices of 2-simplicial neighbours to be stretched to get a chain of length three $1 \circ 1 \circ 1$ from a matrix with a single chain of length two $1 \circ 1$,
- 4 choices of 2-simplicial neighbours to get two chains, one of length 2 the other of length 1, $1 \circ 2$ from a non-symmetric matrix with two chains of length 1, and 2 non-equivalent choices for a symmetric matrix with two chains of length 1,
- one choice to stretch three 2-simplicial neighbours of $\mathbf{D}_3$ to get a matrix with three chains of length 1.

With this type of counting, we get that applying a series of 4 2-simplicial neighbour stretchings to $\mathbf{D}_4$ leads to 52 different 8×8 matrices,

$$\underbrace{2 \cdot [8 + (4 + 4 + 2)]}_{1 \circ [1 \circ 1 \circ 1,\ 1 \circ 2]} + \underbrace{6}_{1 \circ 3} + \underbrace{4 + 4 + (1 + 1^\top)}_{2 \circ 2} = 52$$

applying 5 2-simplicial neighbour stretchings to $\mathbf{D}_4$ leads to 134 different 9×9 matrices,

$$\underbrace{2 \cdot [2 \cdot 18 + 6]}_{1 \circ [1 \circ 1 \circ 1 \circ 1,\ 1 \circ 1 \circ 2,\ 1 \circ 3]} + \underbrace{4 \cdot (4 + 4 + 1) + 2}_{1 \circ 2 \circ 2} + \underbrace{12}_{2 \circ 3} = 134$$

applying 6 leads to 338 different 10×10 matrices

$$\underbrace{2 \cdot [2 \cdot 42 + 38]}_{1 \circ [1 \circ 1 \circ 1 \circ 1 \circ 1, 1 \circ 1 \circ 1 \circ 2, 1 \circ 1 \circ 3, 1 \circ 2 \circ 2]} + \underbrace{4 \cdot 12}_{1 \circ 2 \circ 3} + \underbrace{4(4 + 4 + 1) + 1 + 1^\top}_{2 \circ 2 \circ 2} + \underbrace{8}_{3 \circ 3} = 338$$

and applying 7 leads to 830 different 11 × 11 matrices.

$$\underbrace{2 \cdot [244 + 48]}_{1\circ[1\circ1\circ1\circ1\circ1\circ1,1\circ1\circ1\circ1\circ2,1\circ1\circ1\circ3,1\circ1\circ2\circ2,1\circ2\circ3]} + \underbrace{4 \cdot 37 + 2}_{1\circ2\circ2\circ2} + \underbrace{6 \cdot 8}_{1\circ3\circ3} + \underbrace{4 \cdot 12}_{2\circ2\circ3} = 830$$

This counting shows that there is an infinite zoo of minimally non-firm matrices all containing the single mnsf matrix $\mathbf{D}_4$!

The mnsf matrix $\mathbf{T}_5$,

$$\mathbf{T}_5 = \begin{bmatrix} 1 & 1 & & & \\ 1 & 1 & 1 & & 1 \\ & 1 & 1 & 1 & \\ & & 1 & & 1 \\ & 1 & & 1 & \end{bmatrix},$$

has a very similar structure to $\mathbf{D}_4$ if not even simpler. It has three simplicial 1s at $(1,1), (4,5), (5,4)$ and six 2-simplicial neighbours at $(1,2), (2,1), (2,5), (3,4), (4,3)$ and $(5,2)$. Any matrix that is obtained from $\mathbf{T}_5$ by a series of 2-simplicial neighbour stretchings consists of at most three chains of stretchings. A proof that is structurally identical to the proof of Theorem 5.5.3 shows that stretching the simplicial 1s of any matrix that is obtained from $\mathbf{T}_5$ by a series of 2-simplicial neighbour stretching gives an mnf matrix. Therefore, we get another infinite class of totally balanced mnf matrices, all originating from $\mathbf{T}_5$.

Theorem 5.5.4. *Let $\mathbf{X} \in \{0,1\}^{n \times n}$ be obtained by a series of 2-simplicial neighbour stretchings from $\mathbf{T}_5$ and let Q contain the indices of simplicial 1s of $\mathbf{X}$. Then $\mathcal{S}^Q(\mathbf{X})$ is a minimally non-firm binary matrix.*

Finally let us note, that since one may create a non-firm matrix from every mnsf matrix $\mathbf{X}$ by stretching all 1s of $\mathbf{X}$ that do not belong to an odd hole of $\mathbf{X}$, and every non-firm matrix must contain an mnf matrix, each mnsf matrix can be used to derive at least one mnf matrix. The theorems in this section show that from a single mnsf matrix one may also be able to derive an infinite family of mnf matrices. Therefore, there are much more mnf matrices than mnsf ones.

5.6 Further mnf matrices

In this section, we present some mnf matrices that we discovered computationally and cannot be obtained by Theorem 5.3.1 because they have no simplicial 1s.

Let $\mathbf{\Theta}_n$ be the $n \times n$ matrix obtained from the $n \times n$ mnf matrix $\mathcal{S}^{(n-1,n-1)}(\mathbf{M}_{n-1})$ by turning the 0 at $(n, 1)$ into a 1,

$$\mathbf{\Theta}_n = \begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & 1 & & \\ 1 & & & & 1 & 1 & \\ & & & & 1 & 1 & 1 \\ 1 & & & & & 1 & 1 \end{bmatrix}$$

Observe that $\mathbf{\Theta}_n$ contains two mnsf matrices, one is $\mathbf{M}_{n-1}$ and the other is $\mathbf{H}_3$ in submatrix $\{n-2, n-1, n\} \times \{1, n-1, n-2, n\}$.

Theorem 5.6.1. *$\mathbf{\Theta}_n$ is mnf for all $n \geq 5$.*

Proof. We present a direct proof. To see that $\mathbf{\Theta}_n$ is full-rank, observe that the generalised binary matrix $\mathbf{\Theta}_n^Q$,

$$\mathbf{\Theta}_n^Q = \begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & 1 & & \\ 1 & & & & ? & 1 & \\ & & & & 1 & ? & 1 \\ ? & & & & & ? & 1 \end{bmatrix}$$

has largest rectangle of size 2 and $2n-1$-many 1s, hence by the bound in Equation (1.3.2), we have

$$\lceil \frac{2n-1}{2} \rceil \leq \mathfrak{br}(\mathbf{\Theta}_n^Q) \leq \mathfrak{br}(\mathbf{\Theta}_n) \leq n.$$

$\mathbf{\Theta}_n$ contains $\mathbf{M}_{n-1}$, so $\mathfrak{i}(\mathbf{\Theta}_n) \geq n-1$. Suppose S is an isolated set of size n of $\mathbf{\Theta}_n$. Then as column n only has two 1s, we must have (a) $(n, n) \in S$ or (b) $(n-1, n) \in S$,

$$(a) \begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & 1 & & \\ 1 & & & & 1 & 1 & \\ & & & & 1 & ? & ? \\ ? & & & & & ? & \bullet \end{bmatrix}, \qquad (b) \begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & 1 & 1 & & \\ 1 & & & & 1 & 1 & \\ & & & & ? & ? & \bullet \\ 1 & & & & & ? & ? \end{bmatrix}.$$

In both cases, S then must contain $n-1$ isolated 1s from $\mathbf{M}_{n-1}^{(n-1,n-1)}$ which is impossible as $\mathbf{M}_{n-1}^{(n-1,n-1)}$ is mnf with $\mathfrak{i}(\mathbf{M}_{n-1}^{(n-1,n-1)}) = n-2$ as shown in Theorem 5.4.3. Therefore, $\mathfrak{i}(\mathbf{\Theta}_n) = n-1$.

Suppose that $\mathbf{\Theta}_n$ is not mnf, and has a smallest proper submatrix $\mathbf{Y}$ indexed by $I \times J$ that is not firm. Then $\mathbf{Y}$ is mnf and we must have $n \in I$ and $1 \in J$, as otherwise $\mathbf{Y}$ is a proper submatrix of the mnf matrix $\mathcal{S}^{(n-1,n-1)}(\mathbf{M}_{n-1})$.

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & & n-3 & n-2 & n-1 & n \\
\hline
1 & 1 & 1 & & & & & \\
2 & & 1 & 1 & & & & \\
 & & & \ddots & \ddots & & & \\
n-3 & & & & 1 & 1 & & \\
n-2 & 1 & & & & 1 & 1 & \\
n-1 & & & & & 1 & 1 & 1 \\
n & 1 & & & & & 1 & 1 \\
\hline
\end{array} \tag{5.6.1}$$

If $I \cap [n-3] = \emptyset$ then $\mathbf{Y}$ is a submatrix of $\mathbf{H}_3$ and firm. So $I \cap [n-3] \neq \emptyset$. Note that rows $1, \ldots, n-3$ have only two 1s and columns $2, \ldots, n-3$ also only have two 1s, so $i \in [n-3] \cap I$ implies $i, i+1 \in J$ and $j \in [2, n-3] \cap J$ implies $j-1, j \in I$. Thus $I \cap [n-3] \neq \emptyset$ can only happen if $[n-3] \subset I$ and $[n-2] \subset J$. So at this point we know that $[n-3] \cup \{n\} \subseteq I$ and $[n-2] \subseteq J$.

Let us look at rows $n-2, n-1$. (a) If $n-2 \notin I$, then $\mathbf{Y}$ is a submatrix of $\mathbf{C}_{n-1}$, perhaps with a repeated column, hence superfirm. (b) If $n-1 \notin I$, then $n \notin J$ (as otherwise column n has a single 1), and thus $\mathbf{Y}$ is a submatrix of firm $\mathbf{M}_{n-1}$. So we must have $n-2, n-1 \in I$.

$$(a)\ \begin{array}{c|ccccccc|}
 & 1 & 2 & & n-3 & n-2 & n-1 & n \\
\hline
1 & 1 & 1 & & & & & \\
2 & & 1 & 1 & & & & \\
 & & & \ddots & \ddots & & & \\
n-3 & & & & 1 & 1 & & \\
n-1 & & & & & 1 & 1 & 1 \\
n & 1 & & & & & 1 & 1 \\
\hline
\end{array}, \quad (b)\ \begin{array}{c|ccccccc|}
 & 1 & 2 & & n-3 & n-2 & n-1 & n \\
\hline
1 & 1 & 1 & & & & & \\
2 & & 1 & 1 & & & & \\
 & & & \ddots & \ddots & & & \\
n-3 & & & & 1 & 1 & & \\
n-2 & 1 & & & & 1 & 1 & \\
n & 1 & & & & & 1 & 1 \\
\hline
\end{array}.$$

At this point every row of $\mathbf{\Theta}_n$ is in I, so we must have at least one column in $[n]$ missing so that $\mathbf{Y}$ is a proper submatrix of $\mathbf{\Theta}_n$. (c) If $n-1 \notin J$, then $\mathbf{Y}$ is linear and superfirm. (d) If $n \notin J$ then $\mathbf{Y}$ is of dimension $n \times (n-1)$ and contains $\mathbf{M}_{n-1}$ hence it has $\mathfrak{i}(\mathbf{Y}) = \mathfrak{br}(\mathbf{Y}) = n - 1$.

$$(c)\ \begin{array}{c|cccccc|}
 & 1 & 2 & & n{-}3 & n{-}2 & n \\
\hline
1 & 1 & 1 & & & & \\
2 & & 1 & 1 & & & \\
 & & & \ddots & \ddots & & \\
n{-}3 & & & & 1 & 1 & \\
n{-}2 & 1 & & & & 1 & \\
n{-}1 & & & & & 1 & 1 \\
n & 1 & & & & & 1 \\
\hline
\end{array}\,, \qquad (d)\ \begin{array}{c|cccccc|}
 & 1 & 2 & & n{-}3 & n{-}2 & n{-}1 \\
\hline
1 & 1 & 1 & & & & \\
2 & & 1 & 1 & & & \\
 & & & \ddots & \ddots & & \\
n{-}3 & & & & 1 & 1 & \\
n{-}2 & 1 & & & & 1 & 1 \\
n{-}1 & & & & & 1 & 1 \\
n & 1 & & & & & 1 \\
\hline
\end{array}\,.$$

Therefore, in any case $\mathbf{Y}$ is not mnf, and $\mathbf{\Theta}_n$ is mnf. □

The proof of the above theorem is not very elegant and it would be interesting to know whether a general method exists to turn 0s of known mnf matrices into 1s and obtain further mnf matrices. We tested some cases computationally and discovered a few further mnf matrices. The proof of their mnf-ness is completely enumerational, so we do not present it.

By our computational results, there are no mnf matrices of size 4 × 5 and there are exactly four 5 × 5 mnf matrices which are $\mathcal{S}^{(4,4)}(\mathbf{M}_4)$, $\mathbf{\Theta}_5$ and the below two matrices,

$$[\mathbf{H}_4^\top, \boldsymbol{e}_3 + \boldsymbol{e}_4] = \begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & 1 \\ 1 & & & 1 & 1 \\ 1 & 1 & 1 & 1 & \end{bmatrix}, \qquad \mathbf{\Upsilon}_5 = \begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ 1 & & 1 & 1 & \\ & & 1 & 1 & 1 \\ 1 & 1 & & 1 & 1 \end{bmatrix}.$$

Observe that $[\mathbf{H}_4^\top, \boldsymbol{e}_3 + \boldsymbol{e}_4]$ is similar in structure to matrix $\overline{\mathbf{I}}'_4$ which can be written as $\overline{\mathbf{I}}'_4 = [\mathbf{H}_3^\top, \boldsymbol{e}_2 + \boldsymbol{e}_3]$. Similarly, from computational tests, we think that we may obtain more mnf matrices from mnf matrices $\mathcal{S}^{(n,n)}(\mathbf{M}_n)$. For instance, apart from $\mathbf{\Theta}_6$, the following three 6 × 6 matrices can obtained from $\mathcal{S}^{(5,5)}(\mathbf{M}_5)$ by turning some 0s to 1s,

$$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ 1 & & & 1 & 1 & \\ & & & 1 & 1 & 1 \\ 1 & 1 & & & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ 1 & & & 1 & 1 & \\ & & & 1 & 1 & 1 \\ & 1 & 1 & & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ 1 & & & 1 & 1 & \\ & & & 1 & 1 & 1 \\ 1 & 1 & 1 & & 1 & 1 \end{bmatrix}.$$

Chapter 6

Conclusion and open questions

In the first part of this thesis we looked at firm matrices, which were introduced by Anna Lubiw in her 1990 paper [77].

In the Introduction 2, we gave a detailed account of the history of firm matrices, rectilinear polygon covering and some important problems related to firm matrices.

In Chapter 3, we presented the work of Lubiw in detail. In particular, we defined rectangle cover graphs, superfirmness and generalised binary matrices. Then we explored the polynomial time algorithm to compute a maximum independent set and minimum clique cover of perfect graphs. Unfortunately, we could not adapt it into an algorithm to compute a minimum rectangle cover and a maximum isolated set of firm matrices. Whether this can be done is one of our open questions. Furthermore, in Section 3.3, we provided proof highlights of the most important classes of known firm matrices and some firmness preserving operations. We suspect that $\mathbf{D}_3$-free row-column clutter matrices have full isolation number and this could give a simpler proof of firmness of $\mathbf{D}_3$-free matrices.

In Chapter 4, we explored how minimally imperfect subgraphs can appear in rectangle cover graphs. We proved that in rectangle cover graphs, odd antiholes cannot appear without odd holes. By this we showed that superfirmness is equivalent to the requirement of not having any odd holes in the rectangle cover graph and forbidding odd antiholes is unnecessary. Then we characterised the necessary and sufficient submatrices for 5-holes to appear in the rectangle cover graph. Furthermore, we showed that P_5-free rectangle cover graphs are perfect. In Section 4.3, we defined simplicial 1s and the stretching operation. We proved that 2-simplicial neighbour stretching preserves firmness and superfirmness and used it to show how one can create matrices with nested holes in their rectangle cover graph. Finally, in Section 4.4 we studied minimally non-superfirm matrices. We presented one interval matrix $\mathbf{D}_4$, one totally balanced matrix $\mathbf{T}_5$, one non-square infinite family $\mathbf{H}_n$, and two further infinite families $\mathbf{M}_n$ and $\mathbf{W}_n$ of minimally non-superfirm matrices. We conjecture that $\mathbf{D}_4$ is the only interval minimally non-superfirm matrix and that every minimally non-superfirm matrix $\mathbf{X} \in \{0,1\}^{m\times n}$ is firm and has dimension $|m-n| \leq 1$.

In Chapter 5, first we defined minimally non-firm matrices and explored some basic properties of them and their smallest examples. We suspect that the dimension of every minimally non-firm matrix $\mathbf{X} \in \{0,1\}^{m\times n}$ also satisfies $|m-n| \leq 1$. Afterwards, in Section 5.3 we proved a general theorem which can be used to create minimally non-firm matrices by stretching a carefully selected subset of 1s of matrices that have an odd hole in their rectangle cover graph. We would be curious to understand if every minimally non-firm matrix that has a simplicial 1 arises via this theorem. Then we applied this theorem to obtain a minimally non-firm matrix from every matrix that was proved to be minimally non-superfirm in the previous chapter. After that we used 2-simplicial neighbour stretching to obtain further infinite families of minimally non-firm matrices from $\mathbf{M}_n$, $\mathbf{D}_4$ and $\mathbf{T}_5$. In the future, we would be interested to further generalise the method of proving matrices to be minimally non-firm.

We state one further research avenue that may be interesting to explore. By Lubiw's results it is known that $\mathbf{D}_3$- and $\mathbf{C}_n$-free matrices have chordal rectangle cover graphs. A graph $G = (V, E)$ is chordal if and only if it has a perfect elimination ordering, where a perfect elimination ordering is an ordering $\{v_1, v_2, \ldots, v_n\}$ of V such that each vertex v_i is a simplicial vertex in the subgraph of G which is obtained by deleting vertices $v_1, \ldots, v_{i-1}$ [41]. Such ordering is useful because it leads to simple polynomial time algorithms for chordal graphs to compute a maximum independent set and minimum clique cover. It would be interesting to explore if there is a larger class of matrices than $\mathbf{D}_3$- and $\mathbf{C}_n$-free matrices for which a minimum rectangle cover and a maximum isolated set could be computed by having some sort of ordering of their simplicial 1s and their maximal rectangles. For instance, while $\mathcal{G}(\mathbf{D}_3)$ is not chordal, $\mathbf{D}_3$ can be factorised by first removing the simplicial 1 at $(1,1)$ and then removing any of its other simplicial 1. Similarly, the minimally non-superfirm matrices $\mathbf{D}_4$ and $\mathbf{T}_5$ can be factorised in such way. Of course, an algorithm based on simplicial 1 elimination would only make sense if we require it to work for all submatrices of the matrix. Hence any such matrix class which could be factorised in this way must be a subset of totally balanced matrices.

Part II

Approximate Binary Matrix Factorisation

Chapter 7

Introduction

In the second part of this thesis we look at rank-k binary matrix factorisation (k-BMF). In k-BMF, we are given an input matrix $\mathbf{X} \in \{0,1,?\}^{m\times n}$ and an input positive integer $k \ll \min\{m,n\}$ and need to compute two binary matrices $\mathbf{A} \in \{0,1\}^{m\times k}$ and $\mathbf{B} \in \{0,1\}^{k\times n}$ whose Boolean matrix product $\mathbf{Z} := \mathbf{A} \circ \mathbf{B}$ is closest to $\mathbf{X}$ in the squared Frobenius norm. Therefore, we aim to solve the minimisation problem

$$\zeta(\mathbf{X},k) = \min\{\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{A} \circ \mathbf{B})\|_F^2 : \mathbf{A} \in \{0,1\}^{m\times k}, \mathbf{B} \in \{0,1\}^{k\times n}\},$$

where $\mathcal{P}_\Omega$ is the projection onto the known entries $\Omega(\mathbf{X}) = \mathrm{supp}_0(\mathbf{X}) \cup \mathrm{supp}_1(\mathbf{X})$ of $\mathbf{X}$ and $\|\cdot\|_F$ is the Frobenius norm.

In the next chapters, we present a comprehensive study on integer programming methods for k-BMF. We examine three integer programs in detail: a compact formulation as introduced briefly in our previous short work [61], the exponential formulation of [74] and a novel exponential formulation. We prove several results about the strength of LP-relaxations of the three formulations and their relative comparison that cannot be found in [61] nor in [74].

Our compact IP uses McCormick envelopes [82] to linearise the quadratic terms coming from the matrix product, leading to polynomially many variables and constraints. We prove that for $k > 2$ the LP relaxation of the compact IP has several fractional vertices with objective value 0, hence provides a weak dual bound. In addition, we argue that our compact IP suffers from permutation symmetry.

Our novel exponential formulation overcomes several of these limitations and of other previous approaches. In particular, it does not suffer from permutation symmetry and it does not rely on heuristically guided pattern mining. Moreover, it has a stronger LP relaxation than the compact IP. On the other hand, this formulation has an exponential number of variables which we tackle using a column generation approach that effectively searches over this exponential space without explicit enumeration, unlike the complete enumeration used for the exponential size model of [74].

In addition, we introduce a new objective function for k-BMF under which the problem becomes computationally easier and we explore the relationship between this new objective function and the original squared Frobenius distance.

Finally, we demonstrate that our proposed solution method is able to prove optimality for smaller datasets, while for larger datasets it provides solutions with better accuracy than the state-of-the-art heuristic methods.

The rest of Part II. is organised as follows. In the rest of this chapter, first, we detail some problems related to k-BMF, then we describe a data science example which motivates the study of k-BMF under Boolean arithmetic. We also give a detailed account of previous works related to k-BMF.

In Chapter 8, we give an integer program for 1-BMF and analyse its LP-relaxation. Then we explore an approximation algorithm and some heuristic methods for it.

In Chapter 9, we detail the three IP formulations for k-BMF and prove several results about their LP-relaxations. In addition, we introduce a new objective function and explore its relation to the original squared Frobenius objective.

In the first part of Chapter 10, we detail a framework based on the large scale optimisation technique of column generation for the solution of our exponential formulation. Then, in the second part of the chapter, we demonstrate the practical applicability of our approach on several artificial and real world datasets.

Finally, we state some future research directions and conclude in Chapter 11.

7.1 Related problems

While rank-k binary matrix factorisation relies on the definition of Boolean rank, exact binary matrix factorisation (exact-BMF), in which we aim to compute a factorisation with minimum inner dimension, and k-BMF, where the factorisation sought has fixed inner dimension k, behave quite differently. In exact-BMF it suffices to consider the maximal rectangles of the input matrix $\mathbf{X}$ to obtain a minimum covering of $\text{supp}_1(\mathbf{X})$. In k-BMF, all rank-1 binary matrices are a potential candidate to be included in an optimal factorisation as entries from $\text{supp}_0(\mathbf{X})$ can also be covered. Hence, we cannot restrict our attention to rectangles of $\mathbf{X}$. Due to this difference, most concepts from the first part of this thesis, that intimately relate to the rectangle structure of $\mathbf{X}$ become not so useful. For instance, there is no clear weak dual problem of k-BMF in the way how the isolation number provides a lower bound on the size of an exact factorisation and there is no clear translation of k-BMF to a classical graph problem like the translation of exact-BMF to the minimum clique cover problem through the rectangle cover graph.

The Boolean rank however, can be used to obtain a simple bound on k-BMF as follows. For all k such that $\mathfrak{br}(\mathbf{X}) \leq k$, $\mathbf{X}$ has an exact factorisation with inner dimension $\mathfrak{br}(\mathbf{X})$, so we have $\zeta(\mathbf{X}, k) = 0$. On the other hand, for $k < \mathfrak{br}(\mathbf{X})$, the rank-$k$ factorisation error $\zeta(\mathbf{X}, k)$ is non-zero and decreases strictly with increasing k as the next lemma shows.

Lemma 7.1.1. *For all* $\mathbf{X} \in \{0, 1, ?\}^{m \times n}$ *and* $k < \mathfrak{br}(\mathbf{X})$*, we have* $\zeta(\mathbf{X}, k+1) < \zeta(\mathbf{X}, k)$*.*

Proof. Let $\mathbf{X} \in \{0, 1, ?\}^{m \times n}$ and $k < \mathfrak{br}(\mathbf{X})$. Let $I_1 \times J_1, \ldots, I_k \times J_k$ correspond to the rectangles used in an optimal rank-k factorisation of $\mathbf{X}$ which has error $\zeta(\mathbf{X}, k) > 0$ as $k < \mathfrak{br}(\mathbf{X})$.

If there is any 1 of $\mathbf{X}$ that is not covered in this optimal rank-k factorisation of $\mathbf{X}$, then covering that 1 with a new rectangle gives a rank-$(k+1)$ factorisation with error $\zeta(\mathbf{X}, k) - 1$.

If all 1s of $\mathbf{X}$ are covered by $I_1 \times J_1, \ldots, I_k \times J_k$, then since $\zeta(\mathbf{X}, k) > 0$, there is at least one 0 of $\mathbf{X}$ erroneously covered, say in column j. Let $J'_\ell = J_\ell \setminus \{j\}$ for $\ell \in [k]$. Then $I_1 \times J'_1, \ldots, I_k \times J'_k$ plus the rectangle of column j give a rank-$(k+1)$ factorisation of $\mathbf{X}$ with error at most $\zeta(\mathbf{X}, k) - 1$.

Therefore, in both cases we have $\zeta(\mathbf{X}, k+1) \leq \zeta(\mathbf{X}, k) - 1$. □

By this strict decrease property we may obtain the following simple bound on the rank-k factorisation error in terms of the Boolean rank,

$$\mathfrak{br}(\mathbf{X}) - k \leq \zeta(\mathbf{X}, k).$$

A problem closely related to k-BMF in which 0s of $\mathbf{X}$ are not allowed to be covered is called *rank-k tiling* [65]. In rank-k tiling, it suffices to consider maximal rectangles of $\mathbf{X}$ and the objective is to pick k maximal rectangles of $\mathbf{X}$ which cover the maximum number of 1s of $\mathbf{X}$,

$$\text{tiling}(\mathbf{X}, k) = \min\{\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Z})\|_F^2 : \mathbf{Z} \in \{0,1\}^{m \times n}, \mathfrak{br}(\mathbf{Z}) \leq k, \mathbf{Z} \leq \mathbf{X}\},$$

where $\mathbf{Z} \leq \mathbf{X}$ is understood element-wise and only evaluated over the known entries of $\mathbf{X}$. From this definition it is clear that an optimal rank-k factorisation always has a factorisation error which is less than or equal to the factorisation error of an optimal k-tiling,

$$\zeta(\mathbf{X}, k) \leq \text{tiling}(\mathbf{X}, k).$$

One can also define different variants of k-BMF depending on the underlying arithmetic used when computing the product of binary matrices. For inputs $\mathbf{X}$ and k, rank-k binary matrix factorisation under standard arithmetic seeks to find two binary matrices $\mathbf{A}$ and $\mathbf{B}$ so that the standard matrix product of $\mathbf{A}$ and $\mathbf{B}$ (simply denoted by $\mathbf{AB}$) has integer rank at most k and is closest to $\mathbf{X}$ in the squared Frobenius norm. Recall from Section 1.1.2, that the integer rank, which is also called the rectangle partition number, is the minimum number of disjoint rectangles needed to cover the 1s of a matrix. Hence, in k-BMF under standard arithmetic, every entry of the input matrix may be covered by at most one rank-1 binary matrix. Rank-k BMF under standard arithmetic then can be written as the minimisation problem,

$$\zeta_{\mathbb{Z}}(\mathbf{X}, k) = \min\{\|\mathcal{P}_{\Omega}(\mathbf{X} - \mathbf{AB})\|_F^2 : \mathbf{A} \in \{0,1\}^{m \times k}, \mathbf{B} \in \{0,1\}^{k \times n}\}.$$

Since a rank-k factorisation using standard arithmetic is always a feasible factorisation under Boolean arithmetic, for all $\mathbf{X}$ and k we have

$$\zeta(\mathbf{X}, k) \leq \zeta_{\mathbb{Z}}(\mathbf{X}, k).$$

We mention that in some cases, rank-k BMF under modulo 2 arithmetic [65] is also considered. This arithmetic however does not have a clear relationship to k-BMF under Boolean arithmetic.

7.2 Motivation

Our motivation for rank-k binary matrix factorisation comes from data science applications. Let $\mathbf{X}$ be an $m \times n$ data matrix whose n columns correspond to n features, attributes or observed variables and rows to m data points or observations. Data matrices in practice tend to be very high dimensional and they contain data on a large number of features (columns) in comparison to the number of observations (rows). Low-rank real matrix approximation is an essential tool for dimensionality reduction which helps understand the data better by exposing hidden features. Rank-k real matrix approximation expresses each observation $\mathbf{X}_{i,:}$ as the linear combination of k hidden features $\mathbf{B}_{\ell,:} \in \mathbb{R}^{1 \times n}$ ($\ell \in [k]$),

$$\mathbf{X}_{i,:} \approx a_{i,1}\mathbf{B}_{1,:} + a_{i,2}\mathbf{B}_{2,:} + \cdots + a_{i,k}\mathbf{B}_{k,:} = \sum_{\ell=1}^{k} a_{i,\ell}\mathbf{B}_{\ell,:},$$

where $a_{i,\ell}$ are the real coefficients in the linear combination. In matrix form, rank-k real matrix approximation can be written as $\mathbf{X} \approx \mathbf{AB}$ where the rows of the right factor matrix $\mathbf{B}$ correspond to the k hidden features and the left factor matrix $\mathbf{A}$ contains the coefficients how each observation can be best expressed as a linear combination of the hidden features.

Classical methods for low-rank matrix approximation are not guaranteed to preserve non-negativity of the input matrix $\mathbf{X}$, that is the factor matrices $\mathbf{A}$ and $\mathbf{B}$ can have negative entries even if all the entries of the input matrix $\mathbf{X}$ are non-negative. Non-negative Matrix Factorisation (NMF) addresses this issue, by adding non-negativity constraints on the factor matrices $\mathbf{A}$ and $\mathbf{B}$.

Many practical datasets however, contain categorical features that can only be represented by binary data matrices. In this case, it is natural to require that factor matrices $\mathbf{A}$ and $\mathbf{B}$ to be binary as well. This leads to the problem of rank-k binary matrix factorisation under Boolean arithmetic. For $\mathbf{X} \in \{0,1\}^{m\times n}$, each data point $\mathbf{X}_{i,:}$ is expressed as the Boolean combination of k hidden binary features $\mathbf{B}_{\ell,:} \in \mathbb{R}^{1\times n}$ ($\ell \in [k]$),

$$\mathbf{X}_{i,:} \approx a_{i,1}\mathbf{B}_{1,:} \vee a_{i,2}\mathbf{B}_{2,:} \vee \cdots \vee a_{i,k}\mathbf{B}_{k,:} = \bigvee_{\ell=1}^{k} a_{i,\ell}\mathbf{B}_{\ell,:},$$

where $a_{i,\ell}$ are the binary coefficients in the Boolean combination.

Let us illustrate the interpretability of k-BMF under Boolean arithmetic against k-BMF under standard arithmetic and real and non-negative rank-k factorisation. Consider the data matrix $\mathbf{X}$ (inspired by [84]),

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

where rows correspond to three patients and columns to three symptoms, $x_{ij} = 1$ indicating patient i presents symptom j. The optimal 2-BMF under Boolean arithmetic is given by

$$\mathbf{X} = \mathbf{A} \circ \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

This factorisation describes $\mathbf{X}$ exactly using 2 derived features where the rows of $\mathbf{B}$ specify how the original features relate to the 2 derived features, and the rows of $\mathbf{A}$ give the derived features of each patient. In other words, factor matrix $\mathbf{B}$ reveals that there are 2 underlying diseases that cause the observed symptoms: Disease α is causing symptoms 1 and 2, and disease β is causing symptoms 2 and 3. Matrix $\mathbf{A}$ reveals that patient 1 has disease α, patient 3 has β and patient 2 has both.

Using Boolean arithmetic allows each data point (rows of $\mathbf{X}$) to be expressed as the union of k possibly overlapping derived features. On the contrary if standard arithmetic is used, overlaps are not possible and each data point is the union of k disjoint derived features. Hence, in general one needs larger values of k to achieve accurate factorisations using standard arithmetic. For instance, the below rank-2 factorisation is one of the four optimal solutions with error 1 to 2-BMF under standard arithmetic,

$$\mathbf{X} \approx \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In this factorisation of $\mathbf{X}$, each symptom is caused by exactly one underlying disease, while in real life it is natural to assume that observed symptoms e.g. fever, can be caused by several underlying diseases. The transpose of this factorisation gives another optimal 2-BMF under standard arithmetic, and there are two more optimal factorisations.

$\mathbf{X}$ could also be treated as a real or nonnegative matrix. The best rank-2 real matrix approximation of $\mathbf{X}$ is given by

$$\mathbf{X} \approx \begin{bmatrix} 1.21 & 0.71 \\ 1.21 & 0.00 \\ 1.21 & -0.71 \end{bmatrix} \begin{bmatrix} 0.00 & 0.71 & 0.50 \\ 0.71 & 0.00 & -0.71 \end{bmatrix},$$

while the best rank-2 nonnegative matrix factorisation of $\mathbf{X}$ is given by

$$\mathbf{X} \approx \begin{bmatrix} 1.36 & 0.09 \\ 1.05 & 1.02 \\ 0.13 & 1.34 \end{bmatrix} \begin{bmatrix} 0.80 & 0.58 & 0.01 \\ 0.00 & 0.57 & 0.81 \end{bmatrix}.$$

As neither of these rank-2 approximations are binary, it is harder to find a clear interpretation of them. The rank-2 NMF of $\mathbf{X}$ suggests that symptom 2 presents with lower intensity in both α and β, which is an erroneous conclusion caused by patient 2 that could not have been deduced from data $\mathbf{X}$ which only records presence or absence of symptoms.

As our toy example shows, k-BMF provides interpretable hidden features in some healthcare applications. In addition, BMF derived features of data have also been shown to be interpretable in biclustering gene expression datasets [108], role based access control [74, 75] and market basket data clustering [68].

7.3 Background and previous work

Due to the hardness results of computing any binary matrix factorisation, the majority of methods developed for BMF rely on heuristics. The earliest heuristic in the context of BMF is called Proximus from 2003 by Koyuturk et al. [63]. Proximus aims to factorise a binary matrix under standard arithmetic by recursively partitioning the matrix into submatrices and heuristically computing a rank-1 BMF at each step. To compute a 1-BMF at each step, Koyuturk et al. use an alternating iterative heuristic from a random start which relies on the observation that if $\mathbf{a} \in \{0,1\}^m$ is fixed then $\boldsymbol{b} \in \{0,1\}^n$ which minimises $\|\mathbf{X} - \boldsymbol{a}\boldsymbol{b}^\top\|_F^2$ can be computed in $\mathcal{O}(mn)$ time.

While Proximus is not a heuristic for exact-BMF nor rank-k BMF, because the factorisation that it outputs can cover 0's of the input matrix and it is not of fixed rank-k, it fuelled research on computing efficient and accurate methods for 1-BMF. Shen et al. [100] proposes an integer program (IP) for 1-BMF and several relaxations of it, one of which leads to a 2-approximation, while Shi et al. [101] provides a rounding based 2-approximation which we will detail in Section 8.3.1. Beckerleg et al. [8] extends the Proximus framework to work on binary matrices with missing entries and uses the formulation of [100] to compute 1-BMF at each partitioning step of Proximus.

The problem of rank-k binary matrix factorisation under Boolean arithmetic was first defined by Miettinen at al. in 2006 [83]. Miettinen et al. noted that k-BMF is NP-hard and initialised the hunt for effective heuristic algorithms for k-BMF by describing a heuristic called ASSO. ASSO is based on an association-rule mining approach. It takes $\mathbf{X} \in \{0,1\}^{m\times n}$, the rank k, and a small parameter $\tau \in (0,1)$ as inputs. As a first step, it builds an 'association' matrix $\tilde{\mathbf{B}} \in \{0,1\}^{n\times n}$ with entries defined as

$$\tilde{b}_{t,j} = \begin{cases} 1 & \text{if } \mathbf{X}_{:,t}^\top \mathbf{X}_{:,j} / \mathbf{X}_{:,t}^\top \mathbf{X}_{:,t} \geq \tau, \\ 0 & \text{otherwise.} \end{cases}$$

The next step of ASSO is to choose k rows of $\mathbf{B}$ that will form matrix $\mathbf{B} \in \{0,1\}^{k\times n}$ in the rank-k factorisation $\mathbf{A} \circ \mathbf{B}$. ASSO greedily selects the candidate rows one-by-one from $\tilde{\mathbf{B}}$ to be added to $\mathbf{B}$ by computing how much error each candidate would reduce. ASSO thus consists of three main steps:

(i) Build the association matrix $\tilde{\mathbf{B}}$.

(ii) For each row $j \in [n]$ of $\tilde{\mathbf{B}}$, compute $f(j) = \|\mathbf{X} - \begin{bmatrix}\mathbf{A} & \boldsymbol{a}^{(j)}\end{bmatrix} \circ \begin{bmatrix}\mathbf{B} \\ \tilde{\mathbf{B}}_{j,:}\end{bmatrix}\|_F^2$, where $\boldsymbol{a}^{(j)}$ is set to minimise $f(j)$.

(iii) Let $\ell = \arg\min_j f(j)$. Add row ℓ of $\tilde{\mathbf{B}}$ to $\mathbf{B}$, and the corresponding column $\boldsymbol{a}^{(\ell)}$ to $\mathbf{A}$. If $\mathbf{B}$ has less than k rows go to (ii), otherwise stop.

Step (i) can be done in $\mathcal{O}(mn^2)$ time. In step (ii), $\mathbf{B}$ is appended by row $\tilde{\mathbf{B}}_{j,:}$, and $\mathbf{A}$ is fixed except for one column which is $\boldsymbol{a}^{(j)}$. To get the optimal column $\boldsymbol{a}^{(j)} \in \{0,1\}^m$ which minimises $f(j)$, we can consider each row of $\mathbf{X}$ separately and optimise

$$f(j,i) = \min_{a_i^{(j)} \in \{0,1\}} \|\mathbf{X}_{i,:} - (\mathbf{A}_{i,:} \circ \mathbf{B} \vee a_i^{(j)} \cdot \tilde{\mathbf{B}}_{j,:})\|_F^2$$

for each row $i \in [m]$ of $\mathbf{X}$ and then get $f(j) = \sum_{i=1}^m f(i,j)$. Thus $f(i,j)$ can be computed in $\mathcal{O}(kn)$ time and $f(j)$ in $\mathcal{O}(kmn)$ time. However, if the matrix product of the already fixed parts of $\mathbf{A}$ and $\mathbf{B}$ is precomputed in at most $\mathcal{O}(kmn)$ time, then each $f(j)$ can be computed in $\mathcal{O}(mn)$, thus step (ii) takes time $\mathcal{O}(kmn + mn^2)$. As step (ii) is executed at most k times, the total cost of ASSO is $\mathcal{O}(kmn^2)$.

In addition to ASSO, Miettienen at al. [84] also observes that if $\mathbf{B} \in \{0,1\}^{k \times n}$ is fixed, one can find the optimal $\mathbf{A} \in \{0,1\}^{m \times k}$ that minimises $\|\mathbf{X} - \mathbf{A} \circ \mathbf{B}\|_F^2$ by solving

$$\min_{\mathbf{A}_{i,:} \in \{0,1\}^k} \|\mathbf{X}_{i,:} - \mathbf{A}_{i,:} \circ \mathbf{B}\|_F^2$$

for each row $i \in [m]$ of $\mathbf{X}$ separately. This shows that for fixed $\mathbf{B}$, the optimal $\mathbf{A}$ can be computed in $\mathcal{O}(2^k kmn)$ time (the time analysis we report here is as in Barahona et al. [5] as Miettien et al. inaccurately report $\mathcal{O}(2^k mn)$ by perhaps forgetting to count the cost of matrix product). Barahona et al. in 2019 [5] further improves the ASSO algorithm by embedding it in several alternating style heuristic. We call this improved version of ASSO as ASSO++. However, sadly, none of the variations of ASSO are currently implemented to handle missing entries in the input matrix.

Another approach based on an alternating style heuristic is explored by Zhang et al. in 2007 [108]. They formulate k-BMF under standard arithmetic as a non-linear unconstrained formulation with penalty terms in the objective for non-binary entries,

$$\min_{a,b} \sum_{i=1}^{m}\sum_{j=1}^{n}(x_{i,j} - \sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j})^2 + \lambda \sum_{i=1}^{m}\sum_{\ell=1}^{k}(a_{i,\ell}^2 - a_{i,\ell})^2 + \lambda \sum_{j=1}^{n}\sum_{\ell=1}^{k}(b_{\ell,j}^2 - b_{\ell,j})^2,$$

where $\lambda > 0$. Then they minimise this objective function in an alternating style via gradient descent, keeping $\mathbf{B}$ fixed and solving for $\mathbf{A}$ and vice-versa and increasing the penalty term λ until a sufficient tolerance is reached to round the variables to binary. This algorithm has been implemented in a public matrix factorisation library called 'PyMF - Python Matrix Factorization Module' [99]. Sadly however, the implementation does not support missing entries in the input matrix.

There have also been some integer programming formulations for BMF. Lu et al. [74] presented a series of integer programs for k-BMF and exact-BMF under Boolean arithmetic in 2008. These IPs have exponentially many variables and constraints and require an explicit enumeration of the 2^n possible binary row vectors for factor matrix $\mathbf{B}$. To tackle the exponential explosion of rows considered, a heuristic row generation using association rule mining and subset enumeration is developed [75]. This exponential size IP for k-BMF will be presented in detail in Section 9.3.

Another exponential size integer program for exact-BMF under Boolean arithmetic is presented in [33]. To solve this exponential size IP the authors use either a precomputed enumeration of maximal rectangles or a branch and price method.

Chapter 8

Rank-1 binary matrix factorisation

In this chapter, we spend some time analysing rank-1 binary matrix factorisation. This investigation is useful for most of the approaches that we present for the rank-k case as they will have a component that can be lead back to the rank-1 case.

While 1-BMF seems to be the simplest case, it turns out that it is much more challenging then expected as it is NP-hard as argued in Section 1.6. Recall that for a given generalised binary matrix $\mathbf{X} \in \{0,1,?\}^{m\times n}$ 1-BMF is formulated as

$$\zeta(\mathbf{X},1) = \min_{\boldsymbol{a}\in\{0,1\}^m, \boldsymbol{b}\in\{0,1\}^n} \|\mathcal{P}_\Omega(\mathbf{X} - \boldsymbol{a}\boldsymbol{b}^\top)\|_F^2.$$

In addition, recall the weight matrix $\boldsymbol{\mathcal{W}}$ from Equation (1.4.3), which has components $\mathcal{W}_{i,j} = 2x_{i,j} - 1$ for $(i,j) \in \Omega(\mathbf{X})$ and $\mathcal{W}_{i,j} = 0$ otherwise. Using the weight matrix, and expanding the objective function as in Equation (1.4.2), 1-BMF can also be formulated as a quadratic program with binary variables,

$$\zeta(\mathbf{X},1) = |\operatorname{supp}_1(\mathbf{X})| - \max_{\boldsymbol{a}\in\{0,1\}^m, \boldsymbol{b}\in\{0,1\}^n} \boldsymbol{a}^\top \boldsymbol{\mathcal{W}} \boldsymbol{b}.$$

This formulation is called a *bipartite* binary quadratic program [54] because the variables can be divided into two parts $\boldsymbol{a}$ and $\boldsymbol{b}$ so that quadratic terms only appear between these two parts.

Some simple bounds are immediate from this formulation. For instance, setting $\boldsymbol{a}$ and $\boldsymbol{b}$ to the all 1s or all 0s vectors, we get that $\zeta(\mathbf{X},1) \leq \min\{|\operatorname{supp}_1(\mathbf{X})|, |\operatorname{supp}_0(\mathbf{X})|\}$. Furthermore, if $\boldsymbol{a}$ and $\boldsymbol{b}$ are set to the row and column indicator vectors of a maximum rectangle of $\mathbf{X}$, then we get $\zeta(\mathbf{X},1) \leq |\operatorname{supp}_1(\mathbf{X})| - \mathfrak{mr}(\mathbf{X})$, where $\mathfrak{mr}(\mathbf{X})$ denotes the cardinality of a maximum rectangle of $\mathbf{X}$.

8.1 An integer linear program for 1-BMF

A popular way to tackle quadratic programs that encode NP-hard combinatorial optimisation problems is to formulate them as integer linear programs and then exploit the use of state-of-the-art integer linear program solvers like CPLEX [23]. In this section, we formulate rank-1 BMF as an Integer linear Program (abbreviated simply as IP). So far we only presented a quadratic formulation, which involves the quadratic terms $a_i b_j$ for binary variables a_i, b_j. Here, we use McCormick envelopes [82] to express the nonlinear relationship $y = ab$ in terms of linear constraints only. The McCormick envelopes for $a, b \in \mathbb{R}$ are four linear inequalities given by

$$\begin{aligned} MC(a,b) = \{y \in \mathbb{R}: \quad a + b - y &\leq 1, \\ -a \qquad + y &\leq 0, \\ - b + y &\leq 0, \\ -y &\leq 0\}. \end{aligned}$$

Observe that if $a, b \in \{0,1\}$, then there is only one point in $MC(a,b)$ which is equal to the product of a and b, so $y \in MC(a,b)$ if and only if $y = ab$. Therefore we may use McCormick envelopes to obtain an exact integer linear program for 1-BMF as below,

$$\begin{aligned} (\text{CIP}_1) \quad \zeta_{\text{CIP}}(\mathbf{X}, 1) = \min_{a,b,y} \quad & \sum_{(i,j)\in \text{supp}_1(\mathbf{X})} (1 - y_{i,j}) + \sum_{(i,j)\in \text{supp}_0(\mathbf{X})} y_{i,j} \\ \text{s.t.} \quad & y_{i,j} \in MC(a_i, b_j) && i \in [m], j \in [n], \\ & a_i, b_j \in \{0,1\} && i \in [m], j \in [n]. \end{aligned}$$

Observe that this formulation only has $\mathcal{O}(mn)$ many constraints and variables, hence we denote it by CIP$_1$ which indicates that this is a Compact IP for $k = 1$ (where "compact" means that the formulation has polynomially many variables and constraints in terms of m, n and k). In the next chapter, a compact formulation also using McCormick envelopes will be presented for rank-k BMF. Some variations of CIP$_1$ have appeared in [101] and [100] and in our short NeurIPS workshop paper [61].

By the McCormick envelopes and the binary constraints on a_i, b_j, for each $(i,j) \in [m] \times [n]$ we have $y_{i,j} = a_i b_j$. Hence, setting $\boldsymbol{a}$ to have components a_i, and $\boldsymbol{b}$ to have components b_j and matrix $\mathbf{Y}$ to have components $y_{i,j}$, $\boldsymbol{a}\boldsymbol{b}^\top$ is equal to $\mathbf{Y}$. Therefore, an optimal solution $\mathbf{Y}$ of CIP$_1$ is a rank-1 completion of $\mathbf{X}$ that minimises the original objective $\|\mathcal{P}_\Omega(\mathbf{X} - \mathbf{Y})\|_F^2$.

While CIP$_1$ has only $\mathcal{O}(mn)$ many variables and constraints, some of them are redundant and can be eliminated without impacting any optimal solution. The first simple modification is to eliminate redundant constraints and variables corresponding to missing entries of $\mathbf{X}$. For any $(i,j) \notin \Omega(\mathbf{X})$ variables $y_{i,j}$ do not appear in the objective function, hence it is sufficient to define $y_{i,j}$ for only $(i,j) \in \Omega(\mathbf{X})$ and then only enforce constraints $y_{i,j} \in MC(a_i, b_j)$ for $(i,j) \in \Omega(\mathbf{X})$. Variables a_i and b_j, can still be used to build the rank-1 completion $\boldsymbol{ab}^\top$ of $\mathbf{X}$, which then has each entry for $(i,j) \in \Omega(\mathbf{X})$ equal to the corresponding variable $y_{i,j}$ and also has a binary value for each missing entry of $\mathbf{X}$ at $(i,j) \notin \Omega(\mathbf{X})$.

In addition, we may further reduce the number of constraints based on the optimal value of the variables that appear in the objective function. The following elimination steps were already observed in [61]. Whenever $(i,j) \in \text{supp}_1(\mathbf{X})$, we are maximising variable $y_{i,j}$, hence in an optimal solution $y_{i,j}$ will always take the value at its upper bound and consequently the lower bounding constraints on $y_{i,j}$ for $(i,j) \in \text{supp}_1(\mathbf{X})$ may be deleted without changing the optimum. Similarly, for $(i,j) \in \text{supp}_0(\mathbf{X})$, variables $y_{i,j}$ are being minimised hence the upper bounding constraints for all $y_{i,j}$ with $(i,j) \in \text{supp}_0(\mathbf{X})$ may be eliminated. After these eliminations, variables a_i and b_j, can still be used to build the optimal rank-1 completion $\boldsymbol{ab}^\top$ of $\mathbf{X}$.

Applying all these changes, we obtain a reduced but equivalent version of CIP_1 with half as many constraints as the original one,

$$\begin{aligned}
\zeta_{\text{CIP}}(\mathbf{X},1) = \min_{a,b,y} \quad & \sum_{(i,j)\in\text{supp}_1(\mathbf{X})} (1-y_{i,j}) + \sum_{(i,j)\in\text{supp}_0(\mathbf{X})} y_{i,j} \\
\text{s.t.} \quad & y_{i,j} \le a_i && (i,j) \in \text{supp}_1(\mathbf{X}), \\
& y_{i,j} \le b_j && (i,j) \in \text{supp}_1(\mathbf{X}), \\
& a_i + b_j - 1 \le y_{i,j} && (i,j) \in \text{supp}_0(\mathbf{X}), \\
& y_{i,j} \ge 0 && (i,j) \in \text{supp}_0(\mathbf{X}), \\
& a_i, b_j \in \{0,1\} && i \in [m], j \in [n].
\end{aligned}$$

8.2 Computational investigation of IP

Formulation CIP_1 can be solved via a general purpose IP solver like CPLEX [23]. In this section, we briefly explore solving the reduced version of CIP_1 via CPLEX on a small test set of matrices. In the next sections of this chapter, we will then aim to explain our experimental observations.

Our testing set consists of nine binary matrices which we divide into three groups depending on the dimensions and Boolean rank of the matrices. We call a matrix $\mathbf{X} \in \{0,1\}^{m\times n}$ *small* if $m \in \{20, 35, 50\}$, $n = 20$ and $\mathfrak{br}(\mathbf{X}) \le 5$, *medium* if $m \in \{50, 75, 100\}$, $n = 50$ and $\mathfrak{br}(\mathbf{X}) \le 20$, and *large* if $m \in \{100, 125, 150\}$, $n = 70$ and $\mathfrak{br}(\mathbf{X}) \le 50$. This grouping results in three test classes, each containing three matrices.

In Table 8.1, we present statistics of solving CIP_1 on these test matrices using CPLEX with a time budget of 3600 seconds and otherwise default settings. For each test class, we report the solution time in seconds (*Time*), the number of cutting planes used by CPLEX (*#Cuts*) and the percentage of these cutting planes that are so called $(0, \frac{1}{2})$-cuts or Chvátal-Gomory cuts (*CG*)[1]. We also report the number of nodes processed (*#N.P.*) in the branch and bound tree and the number of nodes remaining to be processed (*#N.R.*). In addition, we report the objective value of the best primal solution (ζ_{CIP_1}) and percentage of optimality gap between the best dual and primal bounds ($\mathrm{Gap}_{\mathrm{CIP}_1}$). Data shown in Table 8.1 corresponds to the arithmetic mean of the values for the three instances in each test class.

Size	$\mathrm{Gap}_{\mathrm{CIP}_1}$	ζ_{CIP_1}	#Cuts	$(0,\frac{1}{2})$	CG	#N.P.	#N.R.	Time
small	0.0	166	36	99.07	0.93	0	0	0.02
medium	0.0	1170	1524	99.67	0.33	174	0	12.47
large	3.9	2989	7111	99.89	0.11	177	55	3600.09

Table 8.1: Solving CIP_1 via CPLEX

From Table 8.1, we observe that CPLEX uses a large number of cutting planes, of which more than 99% are $(0, \frac{1}{2})$-cuts. Furthermore, for small matrices, CIP_1 is solved to optimality by solely using cutting plane methods and primal heuristics. For medium and large matrices, CPLEX uses thousands of $(0, \frac{1}{2})$-cuts and generates a small branch and bound tree of about 200 nodes. We note, that while the gap for large matrices is only about 4%, for all large instances CPLEX runs out of the time budget of 3600 seconds.

Let us further investigate the behaviour of CPLEX on CIP_1. Let CLP_1 denote the LP relaxation of CIP_1 in which the binary constraints $a_i, b_j \in \{0, 1\}$ are relaxed to $a_i, b_j \in [0, 1]$. In Table 8.2, we report the objective value of the LP relaxation of CIP_1 (ζ_{CLP_1}) and the optimality gap between the LP relaxation and the first primal feasible solution found by CPLEX ($\mathrm{Gap}_{\mathrm{CLP}_1}$). In addition, we report statistics at the root node just before going into branching. Column $\zeta_{\mathrm{CLP}_1+\mathrm{Cuts}}$ shows the objective value at the root node which is obtained by adding cutting planes to the LP relaxation, while column $\mathrm{Gap}_{\mathrm{CLP}_1+Cuts}$ gives the optimality gap at the root node.

The results in Table 8.2 suggest that the LP relaxation provides a good dual bound and achieves a 50% optimality gap reduction in average. Furthermore, we can see that adding cutting planes (of which more than 99% is $(0, \frac{1}{2})$-cuts as observed from Table 8.1) to the LP relaxation is extremely effective with either solving the problem to optimality (for small matrices) or reducing the optimality gap to 2% or 6% (for

[1] $(0, \frac{1}{2})$-cuts are a special version of Chvátal-Gomory cuts. The percentages that we report for Chvátal-Gomory cuts in this section, correspond to Chvátal-Gomory cuts that are not $(0, \frac{1}{2})$-cuts.

Size	ζ_{CLP_1}	$\text{Gap}_{\text{CLP}_1}$	$\zeta_{\text{CLP}_1+\text{Cuts}}$	$\text{Gap}_{\text{CLP}_1+\text{Cuts}}$
small	165	52.72	166	0.00
medium	1009	50.00	1141	1.85
large	2132	50.00	2817	6.06

Table 8.2: LP Relaxation and Root Node Bounds for CIP_1 via CPLEX

medium and large matrices). In the next sections, we try to explain the effectiveness of $(0, \frac{1}{2})$-cuts and the 50% optimality gap reduction by the LP relaxation.

8.2.1 LP relaxation

In order to analyse CIP_1's LP-relaxation, let us first focus on the LP relaxation of the McCormick envelopes. In 1989, Padberg defined the Boolean Quadric Polytope (QP) [92] over a non-empty connected graph $G = (V, E)$ as the convex hull of the McCormick envelopes,

$$\text{QP}^G = \text{conv}\{\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} \in \{0,1\}^{|V|+|E|} : y_{i,j} \in MC(x_i, x_j) \text{ for } (i,j) \in E\}.$$

Is is easy to see that the feasible region of CIP_1 with constraints $y_{i,j} \in MC(a_i, b_j)$ defined for all $(i,j) \in [m] \times [n]$ is $\text{QP}^{K_{m,n}}$ while if these constraints are only over $(i,j) \in \Omega(\mathbf{X})$, then the feasible region of CIP_1 is QP^{G_Ω} with $G_\Omega := ([m],[n],\Omega(\mathbf{X}))$. Hence, any knowledge of the LP-relaxation of QP^G is directly relevant for CLP_1.

The LP relaxation of QP^G is obtained by relaxing the integrality constraints,

$$\text{QP}^G_{\text{LP}} = \text{conv}\{\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} \in [0,1]^{|V|+|E|} : y_{i,j} \in MC(x_i, x_j) \text{ for } (i,j) \in E\}.$$

Padberg proves that QP^G_{LP} is half-integral [92, Theorem 7], that is QP^G_{LP} only has vertices with components in $\{0, \frac{1}{2}, 1\}$. In his proof, he shows that every non-singular square submatrix of the constraint matrix of QP^G_{LP} can be reduced to a block diagonal form, with matrices on the diagonal that are an extension of unbalanced cycle matrices with entries from $\{0, \pm 1\}$ and have determinant ± 2. Since the right hand side vector of the McCormick envelopes is a $\{0,1\}$-vector, this shows that every vertex of QP^G_{LP} is half-integral.

Padberg also introduces some valid inequalities for QP^G_{LP} which cut off the fractional half-integral vertices [92]. Let us sum up inequality $x_i + x_j - y_{i,j} \leq 1$ with inequality $-y_{i,j} \leq 0$; and inequality $-x_i + y_{i,j} \leq 0$ with $-x_j + y_{i,j} \leq 0$ to obtain two valid inequalities for QP^G,

$$x_i + x_j - 2y_{i,j} \leq 1, \tag{$\triangle_{ij}$}$$

$$-x_i - x_j + 2y_{i,j} \leq 0. \tag{$\bigtriangledown_{ij}$}$$

Let C be a chordless cycle of graph $G = (V, E)$ over which QP^G is defined and let E_A be a subset of odd cardinality of $E(C)$ and $E_B = E(C) \setminus E_A$. Furthermore, let the inner vertices of E_A and E_B be defined as

$$V_A = \{j : (i, j), (j, k) \in E_A, i \neq k\},$$
$$V_B = \{j : (i, j), (j, k) \in E_B, i \neq k\}.$$

Summing inequalities $\triangle_{ij}$ over $(i, j) \in E_A$ and $\bigtriangledown_{ij}$ over $(i, j) \in E_B$ and then dividing by 2 we get the following inequality

$$\sum_{i \in V_A} x_i - \sum_{i \in V_B} x_i + \sum_{(i,j) \in E_B} y_{i,j} - \sum_{(i,j) \in E_A} y_{i,j} \leq \frac{|E_A|}{2}.$$

This inequality is valid for both QP^G and $\mathrm{QP}^G_{\mathrm{LP}}$ as it is the sum of the four McCormick envelope inequalities with some positive multipliers. Observe that for any feasible point in QP^G the left hand side of this inequality evaluates to an integer number, while the right hand side is fractional as $|E_A|$ was chosen to be odd. Therefore, we may round down the right hand side, without eliminating any feasible points from QP^G and get the valid inequality,

$$\sum_{i \in V_A} x_i - \sum_{i \in V_B} x_i + \sum_{(i,j) \in E_B} y_{i,j} - \sum_{(i,j) \in E_A} y_{i,j} \leq \left\lfloor \frac{|E_A|}{2} \right\rfloor. \tag{8.2.1}$$

An inequality of this form is called an *odd-cycle inequality* for QP^G [92]. One can check that an odd-cycle inequality corresponding to cycle C, odd edge subset $E_A \subset E(C)$ and $E_B = E(C) \setminus E_A$, cuts off the half-integral fractional vertex of $\mathrm{QP}^G_{\mathrm{LP}}$ which has components

$$x_i = \frac{1}{2} \quad i \in C, \qquad y_{i,j} = \frac{1}{2} \quad (i, j) \in E_B, \qquad y_{i,j} = 0 \quad (i, j) \in E_A.$$

While it is also possible to apply the above derivation of odd-cycle inequalities for cycles of G that have some chords, Padberg shows that an odd-cycle inequality is facet defining for QP^G if and only if the cycle over which it is defined is chordless [92, Theorem 9]. Furthermore, Padberg proves that adding all facet defining odd-cycle inequalities to $\mathrm{QP}^G_{\mathrm{LP}}$ is a complete polyhedral description of QP^G if and only if G itself is a chordless cycle [92, Theorem 9]. The separation problem for odd-cycle inequalities is defined as: given a fractional vertex $\boldsymbol{x}$ of $\mathrm{QP}^G_{\mathrm{LP}}$, find an odd-cycle inequality that is violated by $\boldsymbol{x}$ or assert that no such odd-cycle inequality exists. The separation problem for odd-cycle inequalities can be solved in polynomial time via an algorithm due to Barahona et al. [6] using the relationship between the Boolean quadric polytope and the cut polytope [30].

Observe that the odd cycle inequalities were obtained by first taking a non-negative combination of the four McCormick inequalities so that integer variables had only integer coefficients on the left hand side, and then taking the floor function of the fractional right hand side. In general, any inequality obtained via such reasoning is called a *Chvátal-Gomory (CG) inequality or cut* [89, pg. 210]. Furthermore if the non-negative combination uses only 0 and $\frac{1}{2}$ as coefficients then CG-cuts are called $(0, \frac{1}{2})$-*cuts or inequalities* [13]. Observe that in the derivation of the odd cycle inequalities we only used $(0, \frac{1}{2})$-coefficients, hence odd-cycle inequalities are an example of $(0, \frac{1}{2})$-cuts. In [10, Theorem 2 and 3] it is proved that all non-dominated CG-cuts for QP^G are odd-cycle inequalities (where a *non-dominated* CG-cut vaguely means that it cannot be obtained as the sum of some other CG-cuts). Since odd-cycle inequalities defined over chordless cycles of G are facet defining, this shows that adding $(0, \frac{1}{2})$-cuts to $\mathrm{QP}^G_{\mathrm{LP}}$ can lead to a strong relaxation of QP^G.

CLP$_1$. Let us now turn to look at the feasible region of CLP$_1$, the LP relaxation of CIP$_1$. This may be $\mathrm{QP}^{K_{m,n}}_{\mathrm{LP}}$ or $\mathrm{QP}^{G_\Omega}_{\mathrm{LP}}$ depending on whether our input matrix $\mathbf{X}$ has some missing entries and constraints $y_{i,j} \in MC(a_i, b_j)$ are declared for all $(i,j) \in [m] \times [n]$ or just for $(i,j) \in \Omega(\mathbf{X})$. Since, in both of these cases the graph is a bipartite graph, these polytopes are an example of the bipartite Boolean quadric polytope BQP^G where G is a bipartite graph $G = ([m], [n], E)$. For BQP^G, variables x_i can be divided into the two groups a_i, b_j and we can write

$$\mathrm{BQP}^G = \mathrm{conv}\{ \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{y} \end{bmatrix} \in \{0,1\}^{m+n+|E|} : y_{i,j} \in MC(a_i, b_j) \text{ for } (i,j) \in E\}.$$

All of Padberg's results of QP^G immediately apply to BQP^G. Hence, CLP$_1$ has half-integral vertices and for each chordless cycle of $K_{m,n}$ or $G_\Omega = ([m], [n], \Omega(\mathbf{X}))$, several odd-cycle inequalities can be derived which are facet defining for the feasible region of CIP$_1$. Since non-dominated $(0, \frac{1}{2})$-cuts over BQP^G correspond to odd-cycle inequalities, we get an explanation to some degree of why CPLEX is using $(0, \frac{1}{2})$-cuts so successfully at the root node when solving CIP$_1$.

So how do the odd-cycle inequalities for $\mathrm{BQP}^{K_{m,n}}$ look like? The only chordless cycles of $K_{m,n}$ are of size 4, so any facet defining odd-cycle inequality for $\mathrm{BQP}^{K_{m,n}}$ must use a cycle of $K_{m,n}$ with vertices $C = \{i_1, j_1, i_2, j_2\}$. Let the odd edge subset $E_A \subset E(C)$ have $|E_A| = 1$. Then we get the following four odd-cycle inequalities for

C and E_A,

$$\begin{array}{rrrrl}
 & -a_{i_2} & & -b_{j_2} - y_{i_1,j_1} + y_{i_1,j_2} + y_{i_2,j_1} + y_{i_2,j_2} \leq 0 & E_A = \{y_{i_1,j_1}\}, \\
 & -a_{i_2} & -b_{j_1} & + y_{i_1,j_1} - y_{i_1,j_2} + y_{i_2,j_1} + y_{i_2,j_2} \leq 0 & E_A = \{y_{i_1,j_2}\}, \\
-a_{i_1} & & & -b_{j_2} + y_{i_1,j_1} + y_{i_1,j_2} - y_{i_2,j_1} + y_{i_2,j_2} \leq 0 & E_A = \{y_{i_2,j_1}\}, \\
-a_{i_1} & & -b_{j_1} & + y_{i_1,j_1} + y_{i_1,j_2} + y_{i_2,j_1} - y_{i_2,j_2} \leq 0 & E_A = \{y_{i_2,j_2}\}.
\end{array}$$

If $|E_A| = 3$, then the four odd-cycle inequalities for C and E_A are given by

$$\begin{array}{rrrrl}
 & +a_{i_2} & & +b_{j_2} + y_{i_1,j_1} - y_{i_1,j_2} - y_{i_2,j_1} - y_{i_2,j_2} \leq 1 & E_B = \{y_{i_1,j_1}\}, \\
 & +a_{i_2} & +b_{j_1} & - y_{i_1,j_1} + y_{i_1,j_2} - y_{i_2,j_1} - y_{i_2,j_2} \leq 1 & E_B = \{y_{i_1,j_2}\}, \\
+a_{i_1} & & & +b_{j_2} - y_{i_1,j_1} - y_{i_1,j_2} + y_{i_2,j_1} - y_{i_2,j_2} \leq 1 & E_B = \{y_{i_2,j_1}\}, \\
+a_{i_1} & & +b_{j_1} & - y_{i_1,j_1} - y_{i_1,j_2} - y_{i_2,j_1} + y_{i_2,j_2} \leq 1 & E_B = \{y_{i_2,j_2}\}.
\end{array}$$

This shows that one may produce 8 odd-cycle inequalities for every 4-cycle of $K_{m,n}$, of which there are $\binom{m}{2}\binom{n}{2}$ many. Furthermore, if $\mathbf{X}$ has missing entries then there may be more chordless cycles of G_Ω to produce more odd-cycle inequalities according to Equation (8.2.1). However, note that for G_Ω to have a chordless cycle larger than 4, $\mathbf{X}$ must have a square submatrix in which every row and column only has two known entries and all the other entries must be ?s, which is a highly unlikely situation.

In any case, for every 2×2 submatrix of $\mathbf{X}$ which only contains 0s and 1s, 8 odd-cycle inequalities can be added to CLP$_1$ to strengthen the formulation. Padberg's results also tell us that adding all odd-cycle inequalities to CIP$_1$ gives a perfect formulation of 1-BMF if and only if $\mathbf{X}$ has only two known entries in every row and column.

Even if adding all odd-cycles inequalities to CLP$_1$ does not result in an integral feasible region in almost all cases, one may wonder for what matrices $\mathbf{X}$ we can get an objective function that is minimised at an integral vertex of CLP$_1$. Sadly, a submatrix characterisation to such input matrices $\mathbf{X}$ cannot exist as the following example shows.

Example 8.2.1. *Let $\mathbf{P}$ be an arbitrary $p \times q$ binary matrix. Let $\mathbf{X}$ be defined as*

$$\mathbf{X} = \begin{bmatrix} \mathbf{J}_{s,s} & \mathbf{J}_{s,q} \\ \mathbf{J}_{p,s} & \mathbf{P} \end{bmatrix},$$

where $\mathbf{J}_{p,s}$ is the all 1s matrix of dimension $p \times s$. Choosing s so that $s > 2|\operatorname{supp}_0(\mathbf{P})|$, ensures that the optimal rank-1 factorisation to $\mathbf{X}$ is $\mathbf{J}_{(s+p),(s+q)}$ with error $|\operatorname{supp}_0(\mathbf{P})|$. This can be seen as setting $a_i \in \{0, \frac{1}{2}\}$ for any row i or $b_j \in \{0, \frac{1}{2}\}$ for any column j incurs an error greater than $|\operatorname{supp}_0(\mathbf{P})|$.

8.3 Approximation algorithms

Recall that in Section 1.6, we have argued that 1-BMF on $\mathbf{X} \in \{0, 1, ?\}^{m\times n}$ is equivalent to the maximum weight edge biclique problem on the weighted graph $(K_{m,n}, \boldsymbol{\mathcal{W}})$ with weight matrix $\boldsymbol{\mathcal{W}} \in \{0, \pm 1\}^{m\times n}$ defined in Equation (1.4.3) and by this 1-BMF is NP-hard [103, 39]. This equivalence is because we can write

$$\zeta(\mathbf{X}, 1) = |\operatorname{supp}_1(\mathbf{X})| - \max_{\boldsymbol{a}\in\{0,1\}^m, \boldsymbol{b}\in\{0,1\}^n} \boldsymbol{a}^\top \boldsymbol{\mathcal{W}} \boldsymbol{b}.$$

While the optimal solutions of maximum weight edge biclique problem on $(K_{m,n}, \boldsymbol{\mathcal{W}})$ and 1-BMF of $\mathbf{X}$ have an exact correspondence, their suboptimal solutions do not and from an approximation perspective the two problems behave differently.

For instance, the same happens in the case of the maximum independent set and minimum vertex cover problems as pointed out in [38, pg. 133]. $S \subseteq V$ is a maximum independent set of a graph $G = (V, E)$ if and only if $V\backslash S$ is a minimum vertex cover of G. So $\alpha(G) = |V| - \tau(G)$, where $\tau(G)$ is the cardinality of a minimum vertex cover of G. While the minimum vertex cover problem has a polynomial time 2-approximation [38, pg. 134], the maximum independent set problem cannot be approximated in polynomial time to any constant factor unless P=NP (more specifically, approximating the maximum independent set within a factor $\mathcal{O}(|V|^{1-\epsilon})$ is NP-hard[109]).

This reasoning shows that while $\{0, \pm 1\}$-Maximum Weight Edge Biclique problem cannot be approximated in polynomial time within $\mathcal{O}((m + n)^{1-\epsilon})$ for any $\epsilon > 0$ unless P=NP [103, Lemma 4, Theorem 1.], it is possible to have a polynomial time 2-approximation for 1-BMF, which we present in the next section.

8.3.1 2-approximation

In this section we present a polynomial time 2-approximation algorithm for 1-BMF as derived in [101, Theorem 3]. Let us assume that for a given $\mathbf{X} \in \{0, 1, ?\}^{m\times n}$, we have an optimal half-integral LP-relaxation solution $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{y}]$ to CLP$_1$. Then let us define some index sets based on the entries of $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$V_a^{\frac{1}{2}} = \{i \in [m] : a_i = \frac{1}{2}\}, \qquad V_a^1 = \{i \in [m] : a_i = 1\},$$
$$V_b^{\frac{1}{2}} = \{j \in [n] : b_j = \frac{1}{2}\}, \qquad V_b^1 = \{j \in [n] : b_j = 1\}.$$

Observe that if $a_i = 0$ or $b_j = 0$ then $MC(a_i, b_j) = \{0\}$, if $a_i = b_j = \frac{1}{2}$, then $MC(a_i, b_j) = [0, \frac{1}{2}]$ and if $a_i = 1$ then $MC(a_i, b_j) = \{b_j\}$ (or if $b_j = 1$ then

$MC(a_i, b_j) = \{a_i\}$). Therefore, based on the values of a_i and b_j and the objective function, we can easily determine the value of $y_{i,j}$. So let us define some subsets of $\Omega(\mathbf{X})$ as

$$\begin{aligned}
E(V^{\frac{1}{2}}) &= \{(i,j) \in \Omega(\mathbf{X}) : i \in V_a^{\frac{1}{2}}, j \in V_b^{\frac{1}{2}}\},\\
E(V^1) &= \{(i,j) \in \Omega(\mathbf{X}) : i \in V_a^1, j \in V_b^1\},\\
E(V^{\frac{1}{2}} : V^1) &= \{(i,j) \in \Omega(\mathbf{X}) : i \in V_a^{\frac{1}{2}}, j \in V_b^1 \text{ or } i \in V_a^1, j \in V_b^{\frac{1}{2}}\}.
\end{aligned}$$

Using these index sets we may express the objective value of CLP_1 at its optimal solution $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{y}]$ as follows,

$$\begin{aligned}
\zeta_{\text{CLP}}(\mathbf{X}, 1) = {} & |\operatorname{supp}_1(\mathbf{X})| \\
& - |\operatorname{supp}_1(\mathbf{X}) \cap E(V^1)| - \frac{1}{2}|\operatorname{supp}_1(\mathbf{X}) \cap \left[E(V^{\frac{1}{2}}) \cup E(V^{\frac{1}{2}} : V^1)\right]| \\
& + |\operatorname{supp}_0(\mathbf{X}) \cap E(V^1)| + \frac{1}{2}|\operatorname{supp}_0(\mathbf{X}) \cap E(V^{\frac{1}{2}} : V^1)|
\end{aligned}$$

Now let $[\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{y}']$ be the integer point that we obtain by rounding each half component in the optimal solution of CLP_1 to 0. Then the objective function of this integer feasible solution has value

$$\zeta'(\mathbf{X}) = |\operatorname{supp}_1(\mathbf{X})| - |\operatorname{supp}_1(\mathbf{X}) \cap E(V^1)| + |\operatorname{supp}_0(\mathbf{X}) \cap E(V^1)|.$$

Comparing $\zeta'(\mathbf{X})$ to two times the optimal objective value of CLP_1, we get

$$\begin{aligned}
2\zeta_{\text{CLP}}(\mathbf{X}, 1) - \zeta'(\mathbf{X}) \geq {} & |\operatorname{supp}_1(\mathbf{X})| \\
& - |\operatorname{supp}_1(\mathbf{X}) \cap E(V^1)| - |\operatorname{supp}_1(\mathbf{X}) \cap \left[E(V^{\frac{1}{2}}) \cup E(V^{\frac{1}{2}} : V^1)\right]| \\
\geq {} & 0.
\end{aligned}$$

Therefore, this simple rounding gives a 2-approximation for 1-BMF,

$$\zeta_{\text{CIP}}(\mathbf{X}, 1) \leq \zeta'(\mathbf{X}) \leq 2 \cdot \zeta_{\text{CLP}}(\mathbf{X}, 1) \leq 2 \cdot \zeta_{\text{CIP}}(\mathbf{X}, 1).$$

Furthermore, CLP_1 is a compact size linear program, so its optimal solution can be obtained in polynomial time [47], so this LP-based algorithm is a polynomial time 2-approximation for 1-BMF.

The derivation of this 2-approximation, also explains why we see a 50% reduction in the optimality gap when CPLEX solves the LP-relaxation of CIP_1.

Let us understand how useful this 2-approximation is for 1-BMF. Observe that for all matrices $\mathbf{X}$ the following half-integral point is always a feasible solution to CLP_1,

$$a_i = b_j = \frac{1}{2} \qquad (i,j) \in [m] \times [n], \tag{8.3.1}$$

$$y_{i,j} = \frac{1}{2} \qquad (i,j) \in \operatorname{supp}_1(\mathbf{X}), \tag{8.3.2}$$

$$y_{i,j} = 0 \qquad (i,j) \in \operatorname{supp}_0(\mathbf{X}). \tag{8.3.3}$$

We call this half integral point, the *'fully-half'* point. The objective value of CLP_1 at this fully-half point is equal to $\frac{1}{2}|\operatorname{supp}_1(\mathbf{X})|$. Therefore, $\zeta_{\mathrm{CLP}}(\mathbf{X},1) \leq \frac{1}{2}|\operatorname{supp}_1(\mathbf{X})|$ for all matrices $\mathbf{X}$.

Observe that if this fully-half point is an optimal solution of CLP_1 then the simple rounding technique gives the trivial solution of all 0s, which is still a 2-approximation to 1-BMF but completely useless.

In practice, sadly we observe that it is very common that the fully-half point is an optimal solution of CLP_1 and thus the 2-approximation is useless. But this is just an experimental observation and we would be curious to understand the exact conditions under which the fully-half point is an optimal solution to CLP_1. For instance, if $\mathbf{X} \in \{0,1,?\}^{m\times n}$ is very dense, so that it satisfies

$$2\,|\operatorname{supp}_0(\mathbf{X})| < |\operatorname{supp}_1(\mathbf{X})|,$$

then $\boldsymbol{a}\boldsymbol{b}^\top = \mathbf{J}_{m,n}$ gives an error $\zeta_{\mathrm{CIP}}(\mathbf{X},1) = |\operatorname{supp}_0(\mathbf{X})|$ which is less than $\frac{1}{2}|\operatorname{supp}_1(\mathbf{X})|$, so the fully-half point cannot be the optimal solution of CLP_1.

The following proposition gives a characterisation when the fully-half point is optimal for CLP_1.

Proposition 8.3.1. *For $\mathbf{X} \in \{0,1,?\}^{m\times n}$, the fully-half point is an optimal solution of CLP_1 if and only if there exists weights $\gamma_{i,j} \in [0,1]$ for $(i,j) \in \operatorname{supp}_0(\mathbf{X})$ and $\pi_{i,j} \in [0,1]$ for $(i,j) \in \operatorname{supp}_1(\mathbf{X})$ which satisfy the system of equations,*

$$\sum_{j:(i,j)\in\operatorname{supp}_1(\mathbf{X})} \pi_{i,j} + \sum_{j:(i,j)\in\operatorname{supp}_0(\mathbf{X})} \gamma_{i,j} = |\operatorname{supp}_1(\mathbf{X}_{i,:})| \qquad \forall\, i \in [m], \tag{8.3.4}$$

$$\sum_{i:(i,j)\in\operatorname{supp}_1(\mathbf{X})} \pi_{i,j} - \sum_{i:(i,j)\in\operatorname{supp}_0(\mathbf{X})} \gamma_{i,j} = 0 \qquad \forall\, j \in [n]. \tag{8.3.5}$$

Proof. We show that the fully half point and a feasible solution to the dual of CLP_1 is an optimal primal dual pair if and only if the dual variables satisfy Equations (8.3.4) and (8.3.5).

So let us look at the dual of the LP relaxation of the reduced version of CIP_1. Assigning dual variables

- $\mu_{i,j} \geq 0$ to constraints $a_i - y_{i,j} \geq 0$ for $(i,j) \in \operatorname{supp}_1(\mathbf{X})$,
- $\pi_{i,j} \geq 0$ to constraints $b_j - y_{i,j} \geq 0$ for $(i,j) \in \operatorname{supp}_1(\mathbf{X})$,
- $\gamma_{i,j} \geq 0$ to constraints $-a_i - b_j + y_{i,j} \geq -1$ for $(i,j) \in \operatorname{supp}_0(\mathbf{X})$,
- $\alpha_i \geq 0$ to constraints $-a_i \geq -1$ for $i \in [m]$, and

- $\beta_j \geq 0$ to constraints $-b_j \geq -1$ for $j \in [n]$.

we get the following dual program,

$$\begin{aligned}
\max_{\alpha,\beta,\gamma,\mu,\pi} \quad & |\operatorname{supp}_1(\mathbf{X})| - \sum_{i\in[m]} \alpha_i - \sum_{j\in[n]} \beta_j - \sum_{(i,j)\in\operatorname{supp}_0(\mathbf{X})} \gamma_{i,j} \\
\text{s.t.} \quad & \sum_{j:(i,j)\in\operatorname{supp}_1(\mathbf{X})} \mu_{i,j} - \sum_{j:(i,j)\in\operatorname{supp}_0(\mathbf{X})} \gamma_{i,j} \leq \alpha_i & i \in [m] \to (a_i) \\
& \sum_{i:(i,j)\in\operatorname{supp}_1(\mathbf{X})} \pi_{i,j} - \sum_{i:(i,j)\in\operatorname{supp}_0(\mathbf{X})} \gamma_{i,j} \leq \beta_j & j \in [n] \to (b_j) \\
& \mu_{i,j} + \pi_{i,j} \geq 1 & (i,j) \in \operatorname{supp}_1(\mathbf{X}) \to (y_{i,j}) \\
& \gamma_{i,j} \leq 1 & (i,j) \in \operatorname{supp}_0(\mathbf{X}) \to (y_{i,j}) \\
& \alpha_i, \beta_j, \gamma_{i,j}, \mu_{i,j}, \pi_{i,j} \geq 0.
\end{aligned}$$

The fully-half point of CLP$_1$ and a corresponding dual solution are an optimal primal-dual solution pair if and only if they satisfy the complementary slackness conditions. If the complementary slackness conditions hold then

- since $a_i = b_j = \frac{1}{2} < 1$ we must have $\alpha_i = \beta_j = 0$ for all $i \in [m]$ and $j \in [n]$, and
- since $a_i = b_j > 0$ and $y_{i,j} > 0$ for all $(i,j) \in \operatorname{supp}_1(\mathbf{X})$, the first three sets of dual constraints must hold with equality, so we can write $\mu_{i,j} = 1 - \pi_{i,j}$.

Making these simplifications in the constraint set of the dual, we see that there exists a dual solution which satisfies complementary slackness with respect to the fully half point if and only if there exist weights in $[0,1]$ that satisfy Equations (8.3.4) and (8.3.5). □

Using the above proposition, for a small class of 'regular' matrices we are able to show that the fully half point is always an optimal solution of CLP$_1$.

Lemma 8.3.2. *Let* $\mathbf{X} \in \{0,1,?\}^{m\times n}$ *for some* $\delta \geq 1$ *satisfy*

$$\begin{aligned}
|\operatorname{supp}_0(\mathbf{X}_{i,:})| &= \delta |\operatorname{supp}_1(\mathbf{X}_{i,:})| & i \in [m] \\
|\operatorname{supp}_0(\mathbf{X}_{:,j})| &= \delta |\operatorname{supp}_1(\mathbf{X}_{:,j})| & j \in [n].
\end{aligned}$$

Then the 'fully-half' point is an optimal solution of CLP$_1$.

Proof. If $\mathbf{X}$ satisfies the regularity requirement, then by setting $\pi_{i,j} = \frac{1}{2}$ and $\gamma_{i,j} = \frac{1}{2\delta}$ we satisfy the system of equations given in Proposition 8.3.1, and the dual solution

$$\begin{aligned}
\pi_{i,j} &= \frac{1}{2} \quad (i,j) \in \operatorname{supp}_1(\mathbf{X}), & \mu_{i,j} &= \frac{1}{2} \quad (i,j) \in \operatorname{supp}_1(\mathbf{X}), \\
\gamma_{i,j} &= \frac{1}{2\delta} \quad (i,j) \in \operatorname{supp}_0(\mathbf{X}), & \alpha_i &= \beta_j = 0 \quad (i,j) \in [m] \times [n],
\end{aligned}$$

forms an optimal primal dual pair with the fully half point and has objective value $|\operatorname{supp}_1(\mathbf{X})| - 0 - \frac{1}{2\delta}|\operatorname{supp}_0(\mathbf{X})| = \frac{1}{2}|\operatorname{supp}_1(\mathbf{X})|$. □

8.3.2 Greedy algorithm

In this section, we present a greedy algorithm for general bipartite binary quadratic programming (BBQP), which has the form

$$\text{(BBQP)} \qquad \max_{\boldsymbol{a}\in\{0,1\}^m, \boldsymbol{b}\in\{0,1\}^n} \boldsymbol{a}^\top \boldsymbol{\mathcal{H}} \boldsymbol{b}.$$

for some $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{m\times n}$. BBQP can encode the maximum edge biclique problem and the maximum weight edge biclique problem with any weights, hence we cannot expect a constant factor approximation for it. Punnen et al. [54] present several heuristics for BBQP along with a simple but powerful greedy algorithm. The pseudocode of this greedy algorithm is given in Algorithm 3.

Algorithm 3: Greedy Algorithm for BBQP, BBQP($\boldsymbol{\mathcal{H}}$)

```
Input: $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{m\times n}$
Order $i \in [m]$ so that $\sum_{j=1}^n \max(0, \mathcal{H}_{i,j}) \geq \sum_{j=1}^n \max(0, \mathcal{H}_{i+1,j})$.
Set $\boldsymbol{a} = \boldsymbol{0}_m$, $\boldsymbol{s} = \boldsymbol{0}_n^\top$, $\boldsymbol{b} = \boldsymbol{0}_n$.
Phase I.
for $i \in [m]$ do
    $f_0 = \sum_{j=1}^n \max(0, s_j)$
    $f_1 = \sum_{j=1}^n \max(0, s_j + \mathcal{H}_{ij})$
    if $f_0 < f_1$ then
        Set $a_i = 1$, $\boldsymbol{s} = \boldsymbol{s} + \boldsymbol{\mathcal{H}}_{i,:}$
Phase II.
for $j \in [n]$ do
    if $\boldsymbol{s}_j > 0$ then
        Set $b_j = 1$
Output: $\boldsymbol{a} \in \{0,1\}^m, \boldsymbol{b} \in \{0,1\}^n$
```

The essence of Algorithm 3 is to set entries of $\boldsymbol{a}$ and $\boldsymbol{b}$ to 1 which correspond to rows and columns of $\boldsymbol{\mathcal{H}}$ with the largest positive weights. In the first phase of the algorithm, the row indices i of $\boldsymbol{\mathcal{H}}$ are put in decreasing order according to their sum of positive entries. Then sequentially according to this ordering, a_i is set to 1 if $\sum_{j=1}^n \max(0, \sum_{\ell=1}^{i-1} a_\ell \mathcal{H}_{\ell,j}) < \sum_{j=1}^n \max(0, \mathcal{H}_{i,j} + \sum_{\ell=1}^{i-1} a_\ell \mathcal{H}_{\ell,j})$ and 0 otherwise. In the second phase, b_j is set to 1 if $(\boldsymbol{a}^\top \boldsymbol{\mathcal{H}})_j > 0$, and to 0 otherwise. Observe that Phase II. is exactly the algorithm how the optimal solution $\max_{\boldsymbol{b}\in\{0,1\}^n} \boldsymbol{a}^\top \boldsymbol{\mathcal{H}} \boldsymbol{b}$ can be obtained for fixed $\boldsymbol{a}$. Algorithm 3 runs in $\mathcal{O}(mn)$ time. The following result of Punnen et al. show the approximation strength of the greedy algorithm.

Theorem 8.3.3. *[54, Theorem 1.] Let $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{m\times n}$ with $m \leq n$ be an arbitrary input matrix of BBQP. If $m \in \{1, 2\}$, then Algorithm 3 provides the optimal solution of BBQP and if $m > 2$ then Algorithm 3 has an approximation ratio of $m - 1$.*

Proof. Let the rows of $\boldsymbol{\mathcal{H}}$ be ordered so that $\sum_{j=1}^{n} \max(0, \mathcal{H}_{i,j}) \geq \sum_{j=1}^{n} \max(0, \mathcal{H}_{i+1,j})$. In addition, assume that $\sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j}) > 0$, otherwise the problem is trivial. Let $\boldsymbol{a}^*$ and $\boldsymbol{b}^*$ be an optimal solution of BBQP with optimal objective value denoted by $f(\boldsymbol{a}^*, \boldsymbol{b}^*) = \boldsymbol{a}^{*\top}\boldsymbol{\mathcal{H}}\boldsymbol{b}^*$. And let $\boldsymbol{a}'$ and $\boldsymbol{b}'$ be the solution of the greedy algorithm with objective value $f(\boldsymbol{a}', \boldsymbol{b}') = \boldsymbol{a}'^{\top}\boldsymbol{\mathcal{H}}\boldsymbol{b}'$.

If $m = 1$, then the greedy algorithm picks up all positive entries of $\boldsymbol{\mathcal{H}}$ and this solution is optimal. If $m = 2$, then the optimal solution $\boldsymbol{a}^*$ must be one of $\{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\}$. On the other hand, the greedy algorithm selects the best solution from $\{\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\}$. Observe that if $\boldsymbol{a}^* = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$, then $\sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j}) = \sum_{j=1}^{n} \max(0, \mathcal{H}_{2,j})$ must hold, so $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ is also an optimal solution. Therefore, the greedy algorithm picks an optimal solution in this case as well.

Let $m > 2$ and assume that for at least one entry $a_i^* = 0$. Then the optimal objective value satisfies $f(\boldsymbol{a}^*, \boldsymbol{b}^*) \leq (m-1) \cdot \sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j})$, while the greedy objective value satisfies $f(\boldsymbol{a}', \boldsymbol{b}') \geq \sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j})$. Therefore, their ratio is

$$\frac{f(\boldsymbol{a}^*, \boldsymbol{b}^*)}{f(\boldsymbol{a}', \boldsymbol{b}')} \leq \frac{(m-1) \cdot \sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j})}{\sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j})} = m - 1.$$

If $a_i^* = 1$ for all $i \in [m]$, then let $s_i^* = \sum_{j=1}^{n} \mathcal{H}_{i,j} b_j^*$. Then since the greedy algorithm picks the optimal solution for $m = 2$, it must select $a_1' = a_2' = 1$. Thus the greedy objective value satisfies $f(\boldsymbol{a}', \boldsymbol{b}') \geq s_1^* + s_2^*$. Furthermore, for any $i \in [m]$, the greedy objective value also satisfies

$$f(\boldsymbol{a}', \boldsymbol{b}') \geq \sum_{j=1}^{n} \max(0, \mathcal{H}_{1,j}) \geq \sum_{j=1}^{n} \max(0, \mathcal{H}_{i,j}) \geq s_i^*.$$

Using this, the optimal objective value can be bounded as

$$f(\boldsymbol{a}^*, \boldsymbol{b}^*) = \sum_{i=1}^{m} s_i^* \leq (s_1^* + s_2^*) + (m-2) \cdot f(\boldsymbol{a}', \boldsymbol{b}') \leq (m-1) \cdot f(\boldsymbol{a}', \boldsymbol{b}'),$$

which shows that $\frac{f(\boldsymbol{a}^*, \boldsymbol{b}^*)}{f(\boldsymbol{a}', \boldsymbol{b}')} \leq m - 1$. □

The following $m \times m$ matrix is a tight case for the greedy algorithm,

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} 1 & -1 & -1 & \dots & -1 \\ -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \\ -1 & 0 & 0 & & 1 \end{bmatrix} \in \{0, \pm 1\}^{m \times m}.$$

This is because the greedy algorithm only sets $a'_1 = b'_1 = 1$ and the rest of the entries to 0. While the optimal solution has $a^*_i, b^*_j = 1$ for all $i \neq 1$, $j \neq 1$.

We explored this greedy algorithm because it can be used to get good quality heuristic solutions for 1-BMF. In practice, we observe that it provides much more sensible solutions than the CLP_1 based 2-approximation for 1-BMF. Furthermore, this greedy algorithm and several modifications that we detail below, will be used to find rank-1 binary matrices with negative reduced cost as part of a column generation algorithm that we will present in Section 10.1.

There are many variants of Algorithm 3 one can explore. First, the solution greatly depends on the ordering of i's in the first phase. If for some $i_1 \neq i_2$ we have $\sum_{j=1}^n \max(0, \mathcal{H}_{i_1,j}) = \sum_{j=1}^n \max(0, \mathcal{H}_{i_2,j})$, comparing the sum of negative entries of rows i_1 and i_2 can put more "influential" rows of $\boldsymbol{\mathcal{H}}$ ahead in the ordering. Let us call this ordering the *revised ordering* and the one which only compares the positive sums as the *original ordering.* Another option is to use a completely *random order* of i's or to apply a small perturbation to sums $\sum_{j=1}^n \max(0, \mathcal{H}_{i,j})$ to get a *perturbed* version of the revised or original ordering. None of the above ordering strategies clearly dominates the others in all cases but they are fast to compute hence one can evaluate all five ordering strategies (original, revised, original perturbed, revised perturbed, random) and pick the best one. Second, the Algorithm 3 as presented above first fixes $\boldsymbol{a}$ and then $\boldsymbol{b}$. Changing the order of fixing $\boldsymbol{a}$ and $\boldsymbol{b}$ can yield a different result hence it is best to try for both $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{H}}^\top$. In general, it is recommended to start the first phase on the smaller dimension [54]. Third, the solution from Algorithm 3 may be improved by computing the optimal $\boldsymbol{a}$ with respect to fixed $\boldsymbol{b}$. This idea then can be used to fix $\boldsymbol{a}$ and $\boldsymbol{b}$ in an alternating fashion and stop when no changes occur in either.

Chapter 9

Rank-k binary matrix factorisation

In this chapter, we present three integer programs for k-BMF. Recall that the objective function of k-BMF can be expanded as in Equation (1.4.1) to get

$$\|\mathcal{P}_\Omega(\mathbf{X}-\mathbf{Z})\|_F^2 = |\operatorname{supp}_1(\mathbf{X})| - \sum_{(i,j)\in\operatorname{supp}_1(\mathbf{X})} z_{i,j} + \sum_{(i,j)\in\operatorname{supp}_0(\mathbf{X})} z_{i,j}. \tag{9.0.1}$$

Furthermore, observe that Boolean matrix multiplication $\mathbf{Z} = \mathbf{A} \circ \mathbf{B}$ can be written to have entries $z_{i,j} = \min\{1, \sum_\ell a_{i,\ell} b_{\ell,j}\}$ using standard arithmetic summation. In the following sections, we use the linear objective function (9.0.1) and this observation on Boolean matrix products to model k-BMF.

9.1 Compact formulation

We start with a formulation that uses a polynomial number of variables and constraints and has previously appeared in [61]. The following Compact Integer linear Program (CIP) models the entries of matrices $\mathbf{A}, \mathbf{B}, \mathbf{Z}$ directly via binary variables $a_{i,\ell}$, $b_{\ell,j}$ and $z_{i,j}$ respectively and uses McCormick envelopes to avoid the appearance of quadratic terms that would correspond to the constraints $y_{i,\ell,j} = a_{i,\ell} b_{\ell,j}$,

$$\zeta_{\mathrm{CIP}}(\mathbf{X}, k) = \min_{a,b,y,z} \sum_{(i,j)\in\operatorname{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{(i,j)\in\operatorname{supp}_0(\mathbf{X})} z_{i,j} \tag{9.1.1}$$

$$\text{s.t. } y_{i,\ell,j} \le z_{i,j} \le \sum_{l=1}^{k} y_{i,l,j} \qquad i \in [m], j \in [n], \ell \in [k] \tag{9.1.2}$$

$$y_{i,\ell,j} \in MC(a_{i,\ell}, b_{\ell,j}) \qquad i \in [m], j \in [n], \ell \in [k], \tag{9.1.3}$$

$$a_{i,\ell}, b_{\ell,j}, z_{i,j} \in \{0,1\} \qquad i \in [m], j \in [n], \ell \in [k]. \tag{9.1.4}$$

Constraints (9.1.2) encode Boolean matrix multiplication, while a simple modification of the model in which constraints (9.1.2) are replaced by $z_{i,j} = \sum_{\ell=1}^{k} y_{i,\ell,j}$

models k-BMF under standard arithmetic. The McCormick envelopes in constraints (9.1.3) ensure that for $a_{i,\ell}, b_{\ell,j} \in \{0,1\}$, $y_{i,\ell,j}$ are binary variables taking the value $a_{i,\ell}b_{\ell,j}$. Due to the objective function, constraints (9.1.2) and the binary nature of $y_{i,\ell,j}$, the binary constraints on variables $z_{i,j}$ may be relaxed to $z_{i,j} \in [0,1]$ without altering optimal solutions of the formulation.

CIP can easily be adapted to give a polynomial size IP for exact-BMF as follows. Let $t = \min\{m,n\}$. As the Boolean rank is bounded by t, we can replace k in CIP by t. Delete variables $z_{i,j}$ from the model and in constraints (9.1.2) replace $z_{i,j}$ by the input values $x_{i,j}$. Introduce indicator variables $d_\ell \in \{0,1\}$ ($\ell \in [t]$) and add the constraints $a_{i,\ell} \leq d_\ell$ ($i \in [m], \ell \in [t]$) and $b_{\ell,j} \leq d_\ell$ ($j \in [n], \ell \in [t]$). The objective function $\min_{a,b,y,d} \sum_{\ell \in [t]} d_\ell$ then corresponds to minimising the Boolean rank.

The LP relaxation of CIP (CLP) is obtained by replacing constraints (9.1.4) by $a_{i,\ell}, b_{\ell,j}, z_{i,j} \in [0,1]$. For $k=1$, we have $z_{i,j} = y_{i,1,j}$ and and the model reduces to CIP$_1$ from the previous chapter and we know that its LP relaxation CLP$_1$ gives a 2-approximation. This however, does not apply for $k>1$. We next show that CLP for $k>1$ has an objective function value 0.

Proposition 9.1.1. *For any* $\mathbf{X} \in \{0,1,?\}^{n\times m}$ *we have* $\zeta_{\text{CLP}}(\mathbf{X},k) = 0$ *for all* $k>1$. *Moreover, for* $k>2$ CLP *has at least* $k \cdot |\text{supp}_1(\mathbf{X})| + 1$ *vertices with objective value* 0.

Proof. For each $(i,j) \in \text{supp}_1(\mathbf{X})$ let $L_{(i,j)} \subseteq [k]$ such that $|L_{(i,j)}| \geq 2$ and consider the point

$$a_{i,\ell} = \frac{1}{2} \quad i \in [m], \ell \in [k], \qquad\qquad b_{\ell,j} = \frac{1}{2} \quad \ell \in [k], j \in [n],$$

$$y_{i,\ell,j} = \begin{cases} \frac{1}{2} & (i,j) \in \text{supp}_1(\mathbf{X}), \ell \in L_{(i,j)} \\ 0 & \text{otherwise,} \end{cases} \qquad z_{i,j} = \begin{cases} 1 & (i,j) \in \text{supp}_1(\mathbf{X}), \\ 0 & \text{otherwise.} \end{cases}$$

For all $(i,j) \in [m] \times [n]$ and $\ell \in [k]$, setting $a_{i,\ell} = b_{\ell,j} = \frac{1}{2}$ implies that $y_{i,\ell,j} \in MC(\frac{1}{2},\frac{1}{2}) = [0,\frac{1}{2}]$ and $\sum_{l=1}^{k} y_{i,l,j} \geq 1$ holds for all $(i,j) \in \text{supp}_1(\mathbf{X})$, hence this point gives a feasible solution to CLP with objective value 0. For $k=2$, we can only set $L_{(i,j)} = [2]$ for all $(i,j) \in \text{supp}_1(\mathbf{X})$, hence the above construction leads to a single unique point. For $k>2$ however, as the choice of $L_{(i,j)}$'s is arbitrary, there are many feasible points with objective value 0 of this form. As each of these points can differ at only $k \cdot |\text{supp}_1(\mathbf{X})|$ entries corresponding to entries $y_{i,\ell,j}$ for $(i,j) \in \text{supp}_1(\mathbf{X})$, $\ell \in [k]$, there are at most $k \cdot |\text{supp}_1(\mathbf{X})| + 1$ affinely independent points among them. Next we present $k \cdot |\text{supp}_1(\mathbf{X})| + 1$ affinely independent points of this form. Since the objective value is 0 at these points, they must lie on a face of dimension at least

$k \cdot |\operatorname{supp}_1(\mathbf{X})|$ and this face must have at least $k \cdot |\operatorname{supp}_1(\mathbf{X})| + 1$ vertices of CLP with objective value 0. For each $(i,j)^* \in \operatorname{supp}_1(\mathbf{X})$ and $\ell^* \in [k]$, letting $L_{(i,j)} = [k]$ for all $(i,j) \in \operatorname{supp}_1(\mathbf{X}) \setminus \{(i,j)^*\}$ and $L_{(i,j)^*} = [k] \setminus \{\ell^*\}$ provides $k \cdot |\operatorname{supp}_1(\mathbf{X})|$ different points of the above form. Each such point has exactly one entry $y_{i,\ell,j}$ along the indices $(i,j) \in \operatorname{supp}_1(\mathbf{X}), \ell \in [k]$ which is zero. Hence the matrix whose columns correspond to these $k \cdot |\operatorname{supp}_1(\mathbf{X})|$ points has a square submatrix of the form

$$\frac{1}{2}(\mathbf{J}_{k|\operatorname{supp}_1(\mathbf{X})|} - \mathbf{I}_{k|\operatorname{supp}_1(\mathbf{X})|})$$

corresponding to entries $y_{i,\ell,j}$ for $(i,j) \in \operatorname{supp}_1(\mathbf{X}), \ell \in [k]$, where $\mathbf{J}_t$ is the all 1s matrix of size $t \times t$ and $\mathbf{I}_t$ is the identity matrix of size t. Since matrix $\mathbf{J}_t - \mathbf{I}_t$ is nonsingular, the $k \cdot |\operatorname{supp}_1(\mathbf{X})|$ points are linearly independent. In addition, letting $L_{(i,j)} = [k]$ for all $(i,j) \in \operatorname{supp}_1(\mathbf{X})$ gives an additional point for which $y_{i,\ell,j} = \frac{1}{2}$ for all $(i,j) \in \operatorname{supp}_1(\mathbf{X}), \ell \in [k]$, hence the corresponding part of this point is $\frac{1}{2}\mathbf{1}$. Now subtracting $\frac{1}{2}\mathbf{1}$ from the columns of $\frac{1}{2}(\mathbf{J}_{k|\operatorname{supp}_1(\mathbf{X})|} - \mathbf{I}_{k|\operatorname{supp}_1(\mathbf{X})|})$, we get the nonsingular matrix $-\frac{1}{2}\mathbf{I}_{k|\operatorname{supp}_1(\mathbf{X})|}$, hence the $k \cdot |\operatorname{supp}_1(\mathbf{X})| + 1$ above constructed points are affinely independent. □

The above result suggests that unless the factorisation error is 0 i.e. the input matrix is of Boolean rank less than or equal to k, before improving the LP bound of CIP many fractional vertices need to be cut off. To strengthen the formulation of CIP, valid inequalities may be explored. Especially, some of the fractional points that appear in Proposition 9.1.1 may be cut off by some of the odd-cycle inequalities over the bipartite Boolean Quadric Polytope. However, adding all non-dominated odd-cycle inequalities to CLP is not sufficient to cut off all the fractional points with 0 objective value that appear in Proposition 9.1.1. For instance, take $\mathbf{X}$ to be $\mathbf{I}_4$ and set $k = 3$. As $\mathbf{I}_4$ has Boolean rank 4, no zero error rank-3 factorisation exists. Yet, none of the fractional points that appear in Proposition 9.1.1 are cut off by the odd-cycle inequalities.

Furthermore, for $k > 1$, any feasible rank-k factorisation $\mathbf{A} \circ \mathbf{B}$ and a permutation matrix $\mathbf{P} \in \{0,1\}^{k \times k}$ provide another feasible solution $\mathbf{AP} \circ \mathbf{P}^\top\mathbf{B}$ to CIP with the same objective value. Hence, CIP is highly symmetric for $k > 1$. These properties of CIP make it unlikely to be solved to optimality for $k > 1$ in a reasonable amount of time for a large matrix $\mathbf{X}$, though some symmetries may be broken by enforcing lexicographic ordering of rows of $\mathbf{B}$. For small matrices however, CIP constitutes the first approach to get optimal solutions to k-BMF.

9.2 Exponential formulation I.

Recall that any $m \times n$ Boolean rank-k matrix $\mathbf{X}$ can be equivalently written as the Boolean sum of k rank-1 binary matrices

$$\mathbf{X} = \bigvee_{\ell=1}^{k} \boldsymbol{a}_\ell \boldsymbol{b}_\ell^\top \quad \text{for some} \quad \boldsymbol{a}_\ell \in \{0,1\}^m,\ \boldsymbol{b}_\ell \in \{0,1\}^n,\ \ell \in [k].$$

This suggest to directly look for k rank-1 binary matrices instead of introducing variables for all entries of factor matrices $\mathbf{A}$ and $\mathbf{B}$. The second integer program we detail for k-BMF relies on this approach by considering an implicit enumeration of rank-1 binary matrices. Let $\mathcal{R}^{m,n}$ denote the set of all rank-1 binary matrices of dimension $m \times n$ and let $\mathcal{R}^{m,n}_{(i,j)}$ denote the subset of rank-1 matrices of $\mathcal{R}^{m,n}$ which have the (i,j)-th entry equal to 1,

$$\begin{aligned}\mathcal{R}^{m,n} &= \{\boldsymbol{a}\boldsymbol{b}^\top : \boldsymbol{a} \in \{0,1\}^m, \boldsymbol{b} \in \{0,1\}^n, \boldsymbol{a}, \boldsymbol{b} \neq \mathbf{0}\},\\ \mathcal{R}^{m,n}_{(i,j)} &= \{\boldsymbol{a}\boldsymbol{b}^\top \in \mathcal{R}^{m,n} : a_i = b_j = 1\}.\end{aligned}$$

Introducing a binary variable q_r for each rank-1 matrix r in $\mathcal{R}^{m,n}$ and variables $z_{i,j}$ for $(i,j) \in \Omega(\mathbf{X})$, we obtain the following Master Integer linear Program (MIP),

$$\begin{aligned}
\zeta_{\mathrm{MIP}_F}(\mathbf{X}, k) = \min_{z,q} \quad & \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{(i,j)\in \mathrm{supp}_0(\mathbf{X})} z_{i,j} && && (9.2.1)\\
\text{s.t. } \quad & z_{i,j} \le \sum_{r \in \mathcal{R}^{m,n}_{(i,j)}} q_r && (i,j) \in \mathrm{supp}_1(\mathbf{X}) && (9.2.2)\\
& \sum_{r \in \mathcal{R}^{m,n}_{(i,j)}} q_r \le k\, z_{i,j} && (i,j) \in \mathrm{supp}_0(\mathbf{X}) && (9.2.3)\\
& \sum_{r \in \mathcal{R}^{m,n}} q_r \le k && && (9.2.4)\\
& z_{i,j}, q_r \in \{0,1\} && (i,j) \in \Omega(\mathbf{X}),\ r \in \mathcal{R}^{m,n} && (9.2.5)
\end{aligned}$$

The objective, as before, measures the factorisation error in squared Frobenius norm, and subscript F in MIP_F stands for Frobenius. Constraints (9.2.2) and (9.2.3) enforce Boolean matrix multiplication: $z_{i,j}$ takes value 1 if there is at least one active rank-1 binary matrix that covers entry (i,j), otherwise it takes value 0. Notice, that due to the difference in sign of objective coefficients for variables $z_{i,j}$ with $(i,j) \in \mathrm{supp}_1(\mathbf{X})$ and $(i,j) \in \mathrm{supp}_0(\mathbf{X})$ it is enough to declare constraints (9.2.2) and (9.2.3) for indices $(i,j) \in \mathrm{supp}_1(\mathbf{X})$ and $(i,j) \in \mathrm{supp}_0(\mathbf{X})$ respectively. Constraint (9.2.4) ensures that at most k rank-1 binary matrices are active and hence we get a rank-k factorisation of $\mathbf{X}$. Observe that constraints (9.2.2) together with q_r being binary imply that $z_{i,j}$

automatically takes binary values for $(i, j) \in \text{supp}_1(\mathbf{X})$, and due to the objective function it always takes the value at its upper bound, hence $z_{i,j} \in \{0, 1\}$ may be replaced by $z_{i,j} \leq 1$ for all $(i, j) \in \text{supp}_1(\mathbf{X})$ without altering the optimum. In contrast, $z_{i,j}$ for $(i, j) \in \text{supp}_0(\mathbf{X})$ need to be explicitly declared binary as otherwise, if there are some active rank-1 matrices which cover a zero of $\mathbf{X}$, i.e. $q_r > 0$ for some $r \in \mathcal{R}^{m,n}_{(i,j)}$, $(i, j) \in \text{supp}_0(\mathbf{X})$, then variable $z_{i,j}$ corresponding to that zero takes the possibly fractional value $\frac{1}{k} \sum_{r \in \mathcal{R}^{m,n}_{(i,j)}} q_r$. One can also consider a *strong formulation* of MIP_F with exponentially many constraints, in which constraints (9.2.3) are replaced by $q_r \leq z_{i,j}$ for all $r \in \mathcal{R}^{m,n}_{(i,j)}$ and $(i, j) \in \text{supp}_0(\mathbf{X})$.

The LP relaxation of MIP_F (denoted by MLP_F) is obtained by replacing the integrality constraints by $z_{i,j}, q_r \in [0, 1]$. Unlike CLP, the optimal objective value of MLP_F ($\zeta_{\text{MLP}_F}(\mathbf{X}, k)$) is not always zero. By comparing the rank of the factorisation, k to the isolation number $\mathfrak{i}(\mathbf{X})$ of the input matrix $\mathbf{X}$ we can deduce when MLP_F takes non-zero objective value.

Proposition 9.2.1. *Let $\mathbf{X} \in \{0, 1, ?\}^{m \times n}$ and $k \in \mathbb{Z}_{++}$. If $\mathfrak{i}(\mathbf{X}) > k$, then the optimal objective value of the LP relaxation of MIP_F satisfies*

$$\zeta_{\text{MLP}_F}(\mathbf{X}, k) \geq \frac{1}{k} \left(\mathfrak{i}(\mathbf{X}) - k \right).$$

Proof. Let S be an isolated set of $\mathbf{X}$ of cardinality $\mathfrak{i}(\mathbf{X})$. We will establish a feasible solution to the dual of MLP_F (MDP_F) with objective value $\frac{1}{k} \left(\mathfrak{i}(\mathbf{X}) - k \right)$ implying the result.

Let us apply a change of variables $\xi_{i,j} = 1 - z_{i,j}$ for $(i, j) \in \text{supp}_1(\mathbf{X})$ for the ease of avoiding the constant term in the objective function of MLP_F. Then the bound constraints of MLP_F can be written as $\xi_{i,j} \geq 0$ for $(i, j) \in \text{supp}_1(\mathbf{X})$, $z_{i,j} \geq 0$ for $(i, j) \in \text{supp}_0(\mathbf{X})$ and $q_r \geq 0$, $r \in \mathcal{R}^{m,n}$ as the objective function is minimising both $\xi_{i,j}$ and $z_{i,j}$ and we have the cardinality constrains on q_r. Associating dual variables $p_{i,j} \geq 0$ $(i, j) \in \text{supp}_1(\mathbf{X})$ with constraints $\sum_{r \in \mathcal{R}^{m,n}_{i,j}} q_r + \xi_{i,j} \geq 1$; dual variables $s_{i,j} \geq 0$ $(i, j) \in \text{supp}_0(\mathbf{X})$ with constraints (9.2.3) and dual variable $\mu \geq 0$ with constraint

(9.2.4), the Master Dual Program ($\mathrm{MDP_F}$) of $\mathrm{MLP_F}$ is

$$\begin{aligned}
\zeta_{\mathrm{MDP}_F}(\mathbf{X},k) = & \max_{p,s,\mu} \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} p_{i,j} - k\,\mu \\
& \text{s.t.} \quad \sum_{\substack{(i,j)\in\\ \mathrm{supp}_1(\mathbf{X})\cap \mathrm{supp}_1(\mathbf{R})}} p_{i,j} - \sum_{\substack{(i,j)\in\\ \mathrm{supp}_0(\mathbf{X})\cap \mathrm{supp}_1(\mathbf{R})}} s_{i,j} \le \mu \qquad \mathbf{R}\in\mathcal{R}^{m,n}, \qquad (9.2.6)\\
& \qquad 0 \le p_{,ij} \le 1 \qquad (i,j)\in \mathrm{supp}_1(\mathbf{X}),\\
& \qquad 0 \le s_{i,j} \le \frac{1}{k} \qquad (i,j)\in \mathrm{supp}_0(\mathbf{X}),\\
& \qquad 0 \le \mu.
\end{aligned}$$

Let $s_{i,j} = \frac{1}{k}$ for $(i,j)\in \mathrm{supp}_0(\mathbf{X})$ and let $p_{i,j} = \frac{1}{k}$ for $(i,j)\in S$ and $p_{i,j} = 0$ for all other $(i,j)\in \mathrm{supp}_1(\mathbf{X})\setminus S$. The bound constraints on $p_{i,j}$ and $s_{i,j}$ are satisfied then. It remains to choose $\mu \ge 0$ so that we satisfy constraint (9.2.6) for all rank-1 binary matrices $\mathbf{R}\in\mathcal{R}^{m,n}$. Let $\mathbf{R}\in\mathcal{R}^{m,n}$ correspond to a rectangle of $\mathbf{X}$, so we have $|\mathrm{supp}_0(\mathbf{X})\cap \mathrm{supp}_1(\mathbf{R})| = 0$. Then by the definition of isolated sets, $\mathbf{R}$ can contain at most one element from S and hence we have $|\mathrm{supp}_1(\mathbf{R})\cap S| \le 1$. This tells us that for any $\mu \ge \frac{1}{k}$, constraint (9.2.6) is satisfied for all $\mathbf{R}\in\mathcal{R}^{m,n}$ that corresponds to a rectangle of $\mathbf{X}$. Now let $\mathbf{R}\in\mathcal{R}^{m,n}$ be a rank-1 binary matrix which covers at least one zero entry of $\mathbf{X}$. Then $\mathbf{R}$ may contain more than one element from S. However, if it contains more than one element from S then it must also contain at least $\binom{|\mathrm{supp}_1(\mathbf{R})\cap S|}{2}$-many zeros as for any two distinct elements $(i_1,j_1),(i_2,j_2)$ in S we have $(i_1,j_2)\in \mathrm{supp}_0(\mathbf{X})$ or $(i_2,j_1)\in \mathrm{supp}_0(\mathbf{X})$ by the definition. Hence, for all $\mathbf{R}\in\mathcal{R}^{m,n}$ such that $|\mathrm{supp}_0(\mathbf{X})\cap \mathrm{supp}_1(\mathbf{R})| > 0$, constraint (9.2.6) satisfies

$$\begin{aligned}
&\frac{1}{k}|S\cap \mathrm{supp}_1(\mathbf{R})| - \frac{1}{k}|\mathrm{supp}_0(\mathbf{X})\cap \mathrm{supp}_1(\mathbf{R})|\\
&\le \frac{1}{k}|S\cap \mathrm{supp}_1(\mathbf{R})| - \frac{1}{k}\binom{|S\cap \mathrm{supp}_1(\mathbf{R})|}{2} \le \frac{1}{k}.
\end{aligned}$$

Thus we can set $\mu = \frac{1}{k}$ to get the objective value

$$\frac{1}{k}\left(\mathfrak{i}(\mathbf{X}) - k\right) \le \zeta_{\mathrm{MDP}_F}(\mathbf{X},k) = \zeta_{\mathrm{MLP}_F}(\mathbf{X},k).$$

□

The following example shows that we cannot strengthen Proposition 9.2.1 by replacing the condition $k < \mathfrak{i}(\mathbf{X})$ with the requirement that k has to be strictly smaller than the Boolean rank of $\mathbf{X}$.

Example 9.2.2. *Let* $\mathbf{X} = \bar{\mathbf{I}}_4$*, the* 4×4 *complement identity matrix which has* $\mathfrak{i}(\bar{\mathbf{I}}_4) = 3$ *and* $\mathfrak{br}(\bar{\mathbf{I}}_4) = 4$ *(and is minimally non-firm). Since* $\bar{\mathbf{I}}_4$ *is of Boolean rank* 4*, for* $k = 3$ *we have* $\zeta(\bar{\mathbf{I}}_4, 3) > 0$*. On the other hand, the optimal objective value* $\zeta_{\text{MLP}_F}(\bar{\mathbf{I}}_4, 3)$ *is* 0 *which is attained by a fractional solution in which the following* 6 *rank-*1 *binary matrices are active each with weight* $q_r = \frac{1}{2}$*,*

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

9.3 Exponential formulation II.

For $t \in [2^n - 1]$ let $\boldsymbol{\beta}_t \in \{0, 1\}^n$ be the vector denoting the binary encoding of t and note that these vectors give a complete enumeration of all non-zero binary vectors of size n. Let $\beta_{t,j}$ denote the j-th entry of $\boldsymbol{\beta}_t$. In [74], the authors present the following Exponential size Integer linear Program (EIP) formulation using a separate indicator variable d_t for each one of these exponentially many binary vectors $\boldsymbol{\beta}_t$,

$$\zeta_{\text{EIP}}(\mathbf{X}, k) = \min_{\alpha,z,d} \sum_{(i,j) \in \text{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{(i,j) \in \text{supp}_0(\mathbf{X})} z_{i,j} \tag{9.3.1}$$

$$\text{s.t. } z_{i,j} \leq \sum_{t=1}^{2^n-1} \alpha_{i,t}\beta_{t,j} \qquad (i,j) \in \text{supp}_1(\mathbf{X}), \tag{9.3.2}$$

$$\sum_{t=1}^{2^n-1} \alpha_{i,t}\beta_{tj} \leq k z_{i,j} \qquad (i,j) \in \text{supp}_0(\mathbf{X}) \tag{9.3.3}$$

$$\sum_{t=1}^{2^n-1} d_t \leq k \tag{9.3.4}$$

$$\alpha_{i,t} \leq d_t \qquad i \in [m], t \in [2^n - 1], \tag{9.3.5}$$

$$z_{i,j}, d_t, \alpha_{i,t} \in \{0, 1\} \qquad (i,j) \in \Omega(\mathbf{X}), t \in [2^n - 1]. \tag{9.3.6}$$

The above formulation has an exponential number of variables and constraints but it is an integer linear program as $\beta_{t,j}$ are input parameters to the model. Let ELP be the LP relaxation of EIP. Observe that due to the objective function the bound constraints in ELP may be simplified to $z_{i,j}, \alpha_{i,t}, d_t \geq 0$ for all i, j, t and $z_{i,j} \leq 1$ for $(i, j) \in \text{supp}_1(\mathbf{X})$ without changing the optimum. To solve EIP or ELP explicitly, one needs to enumerate all binary vectors $\boldsymbol{\beta}_t$ ($t \in [2^n - 1]$), which is possible only up to a very limited size. To the best of our knowledge, no method is available that avoids explicit enumeration and can guarantee the optimal solution of EIP. Previous attempts at computing a rank-k factorisation via EIP all relied on working with only a small heuristically chosen subset of vectors $\boldsymbol{\beta}_t$ [74, 75]. However, if there was an efficient method to solve ELP, the following result shows it to be as strong as the LP relaxation of MIP_F.

Proposition 9.3.1. *The optimal objective values of* ELP *and* MLP_F *are equal.*

Proof. Note that due to constraints (9.2.2) and (9.2.3) in MLP_F and constraints (9.3.2) and (9.3.3) in ELP, it suffices to show that for any feasible solution $\alpha_{i,t}, d_t$ of ELP one can build a feasible solution q_r of MLP_F for which $\sum_{t=1}^{2^n-1} \alpha_{i,t}\beta_{t,j} = \sum_{r\in\mathcal{R}^{m,n}_{(i,j)}} q_r$, and vice-versa.

First consider a feasible solution $\boldsymbol{\alpha}_t \in \mathbb{R}^m$, $d_t \in \mathbb{R}$ (for $t \in [2^n - 1]$) to ELP and note that by constraint (9.3.5) we have $0 \leq \alpha_{i,t} \leq d_t$ for all $i \in [m]$ and $t \in [2^n - 1]$. We can therefore express each $\boldsymbol{\alpha}_t$ as a convex combination of binary vectors in $\{0, 1\}^m$ scaled by d_t,

$$\boldsymbol{\alpha}_t = d_t \sum_{s=1}^{2^m-1} \lambda_{s,t}\, \boldsymbol{a}_s \quad \boldsymbol{a}_s \in \{0,1\}^m \setminus \{\mathbf{0}\}, \quad \sum_{s=1}^{2^m-1} \lambda_{s,t} \leq 1, \quad \lambda_{s,t} \geq 0, \quad s \in [2^m - 1]$$

where $\boldsymbol{a}_s$ denotes the binary encoding of s. Note that we do not require $\lambda_{s,t}$'s to add up to 1 as we exclude the zero vector. We can therefore rewrite the solution of ELP as follows

$$\sum_{t=1}^{2^n-1} \boldsymbol{\alpha}_t \boldsymbol{\beta}_t^\top = \sum_{t=1}^{2^n-1} \left(\sum_{s=1}^{2^m-1} d_t\, \lambda_{s,t}\, \boldsymbol{a}_s \right) \boldsymbol{\beta}_t^\top = \sum_{s=1}^{2^m-1} \sum_{t=1}^{2^n-1} q_{s,t} \boldsymbol{a}_s \boldsymbol{\beta}_t^\top \quad \text{where } q_{s,t} := d_t\, \lambda_{s,t}.$$

Now it is easy to see that $\boldsymbol{a}_s\boldsymbol{\beta}_t^\top \in \mathcal{R}^{m,n}$ and since $\sum_{t=1}^{2^n-1} d_t \leq k$ holds in any feasible solution to ELP, we get $\sum_{s=1}^{2^m-1}\sum_{t=1}^{2^n-1} q_{s,t} \leq k$, which shows that $q_{s,t}$ is feasible for MLP_F.

The construction works backwards as well, as any feasible solution to MLP_F can be written as $\sum_{s=1}^{2^m-1}\sum_{t=1}^{2^n-1} q_{s,t}\boldsymbol{a}_s\boldsymbol{\beta}_t^\top$ for some rank-1 binary matrices $\boldsymbol{a}_s\boldsymbol{\beta}_t^\top \in \mathcal{R}^{m,n}$ and corresponding variables $q_{s,t} \geq 0$. Now let $\boldsymbol{\alpha}_t := \sum_{s=1}^{2^m-1} q_{s,t}\, \boldsymbol{a}_s$ and $d_t := \max_{i\in[m]} \alpha_{i,t}$ to satisfy $\alpha_{i,t} \leq d_t$. Then since we started from a feasible solution to MLP_F, we have $\sum_{s=1}^{2^m-1}\sum_{t=1}^{2^n-1} q_{s,t} \leq k$ and hence $\sum_{t=1}^{2^n-1} d_t \leq k$ is satisfied too. □

9.4 Working under a new objective

In the previous section, we presented formulations for k-BMF which measured the factorisation error in the squared Frobenius norm, which coincides with the entry-wise ℓ_1 norm as showed in Equation (9.0.1). In this section, we explore another objective function which introduces an asymmetry between how false negatives and false positives are treated. Whenever a 0 entry is erroneously covered in a rank-k factorisation, it may be covered by up to k rank-1 binary matrices. Our new objective function attributes a weighted error term to each 0 entry which is proportional to the number of rank-1 matrices covering that entry. As previously, by denoting $\mathbf{Z} = \mathbf{A} \circ \mathbf{B}$ a rank-k factorisation of $\mathbf{X}$, the new objective function is

$$\zeta_{(\rho)}(\mathbf{X}, k) = \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \rho \sum_{(i,j)\in \mathrm{supp}_0(\mathbf{X})} \sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j}. \tag{9.4.1}$$

Note that the constraints $a_{i,\ell} b_{\ell,j} \le z_{i,j} \le \sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j}$ encoding Boolean matrix multiplication imply that $\frac{1}{k}\sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j} \le z_{i,j} \le \sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j}$. Therefore, denoting the original squared Frobenius norm objective function in Equation (9.0.1) by $\zeta_F(\mathbf{X}, k)$, for any $\mathbf{X}$ and rank-k factorisation $\mathbf{Z}$ of $\mathbf{X}$ the following relationship holds between $\zeta_F(\mathbf{X}, k)$ and $\zeta_{(1)}(\mathbf{X}, k)$,

$$\zeta_F(\mathbf{X}, k) \le \zeta_{(1)}(\mathbf{X}, k) \le \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{(i,j)\in \mathrm{supp}_0(\mathbf{X})} k\, z_{i,j} \le k \cdot \zeta_F(\mathbf{X}, k)$$

and between $\zeta_F(\mathbf{X}, k)$ and $\zeta_{(\frac{1}{k})}(\mathbf{X}, k)$,

$$\frac{1}{k} \cdot \zeta_F(\mathbf{X}, k) \le \zeta_{(\frac{1}{k})}(\mathbf{X}, k) \le \zeta_F(\mathbf{X}, k).$$

We next show that this new objective function $\zeta_{(\rho)}(\mathbf{X}, k)$ with $\rho = 1$ can overestimate the original objective $\zeta_F(\mathbf{X}, k)$ by a factor of k. But first, we need a technical result which shows that whenever the input matrix $\mathbf{X}$ contains repeated rows or columns we may assume that an optimal factorisation exists which has the same row-column repetition pattern.

Lemma 9.4.1 (Preprocessing). *Let $\mathbf{X}$ contain some duplicate rows and columns. Then there exists an optimal rank-k binary matrix factorisation of $\mathbf{X}$ under objective $\zeta_F(\mathbf{X}, k)$ (or $\zeta_{(\rho)}(\mathbf{X}, k)$) whose rows and columns corresponding to identical copies in $\mathbf{X}$ are identical.*

Proof. Since the transpose of an optimal rank-k factorisation is optimal for $\mathbf{X}^\top$, it suffices to consider the rows of $\mathbf{X}$. Furthermore, it suffices to consider only one set of repeated rows of $\mathbf{X}$, so let $I \subseteq [m]$ be the index set of a set of identical rows of $\mathbf{X}$. We then need to show that there exists an optimal rank-k factorisation whose rows indexed by I are identical. Let $\mathbf{Z} = \mathbf{A} \circ \mathbf{B}$ be an optimal rank-k factorisation of $\mathbf{X}$ under objective $\zeta_F(\mathbf{X}, k)$. For all $i_1, i_2 \in I$ we must have

$$\sum_{j:(i_1,j)\in \operatorname{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{j:(i_1,j)\in \operatorname{supp}_0(\mathbf{X})} z_{i,j} = \sum_{j:(i_2,j)\in \operatorname{supp}_1(\mathbf{X})} (1 - z_{i,j}) + \sum_{j:(i_2,j)\in \operatorname{supp}_0(\mathbf{X})} z_{i,j}$$

as otherwise replacing $\mathbf{A}_{i,:}$ for each $i \in I$ with row $\mathbf{A}_{i^*,:}$ where $i^* \in I$ is a row index for which the above sum is minimised leads to a smaller error factorisation. Then since the condition stated in the equation above holds, replacing $\mathbf{A}_{i,:}$ for each $i \in I$ with row $\mathbf{A}_{i^*,:}$ for any $i^* \in I$ leads to an optimal solution of the desired property. Similarly, if $\mathbf{Z}$ is an optimal factorisation under objective $\zeta_{(\rho)}(\mathbf{X}, k)$, then for all $i_1, i_2 \in I$ the corresponding objective terms must equal and hence an optimal solution of the desired property exists. □

This result implies that whenever the input matrix $\mathbf{X}$ contains repeated rows or columns we may solve the following problem on a smaller matrix instead. Let $\mathbf{X}' \in \{0,1\}^{m' \times n'}$ be the binary matrix obtained from $\mathbf{X}$ by replacing each duplicate row and column by a single representative and let $\mathfrak{r} \in \mathbb{Z}_+^{m'}$ and $\mathfrak{c} \in \mathbb{Z}_+^{n'}$ be the counts of each unique row and column of $\mathbf{X}'$ in $\mathbf{X}$ respectively. By Lemma 9.4.1 an optimal rank-k factorisation $\mathbf{Z}' = \mathbf{A}' \circ \mathbf{B}'$ of $\mathbf{X}'$ under the *updated* objective function

$$\zeta_F(\mathbf{X}', k, \mathfrak{r}, \mathfrak{c}) := \sum_{(i,j)\in \operatorname{supp}_1(\mathbf{X}')} \mathfrak{r}_i\, \mathfrak{c}_j\, (1 - z'_{i,j}) + \sum_{(i,j)\in \operatorname{supp}_0(\mathbf{X}')} \mathfrak{r}_i\, \mathfrak{c}_j\, z'_{i,j}$$

or

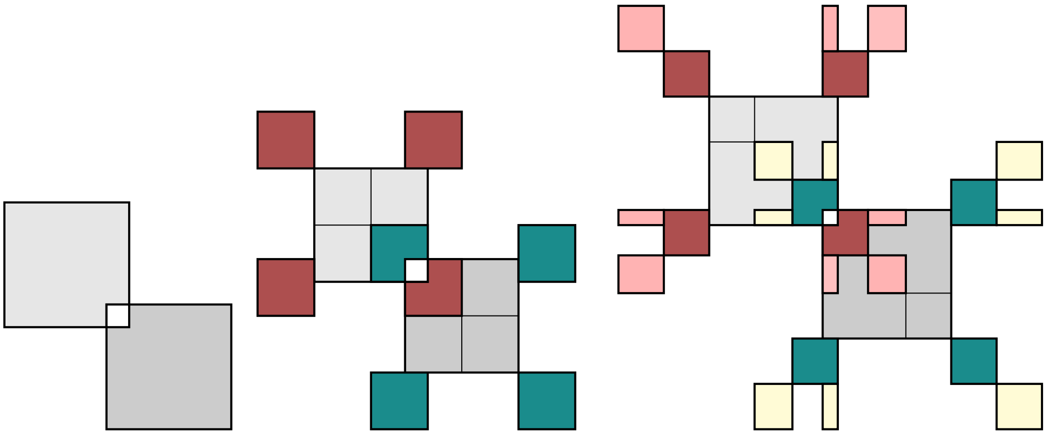
$$\zeta_{(\rho)}(\mathbf{X}', k, \mathfrak{r}, \mathfrak{c}) := \sum_{(i,j)\in \operatorname{supp}_1(\mathbf{X}')} \mathfrak{r}_i\, \mathfrak{c}_j\, (1 - z'_{i,j}) + \rho \sum_{(i,j)\in \operatorname{supp}_0(\mathbf{X}')} \mathfrak{r}_i\, \mathfrak{c}_j\, a'_{i,\ell} b'_{\ell,j}$$

leads to an optimal rank-k factorisation of $\mathbf{X}$ under the original objective function $\zeta_F(\mathbf{X}, k)$ or $\zeta_{(\rho)}(\mathbf{X}, k)$.

Proposition 9.4.2. *For each positive integer k there exists a binary matrix $\mathbf{X}^{(k)}$ for which the optimal rank-k binary matrix factorisations under objectives $\zeta_F(\mathbf{X}^{(k)}, k)$ and $\zeta_{(1)}(\mathbf{X}^{(k)}, k)$ satisfy*

$$\zeta_{(1)}(\mathbf{X}^{(k)}, k) = k \cdot \zeta_F(\mathbf{X}^{(k)}, k).$$

Proof. The idea behind the proof is to consider a matrix $\mathbf{Z}^{(k)}$ of exact Boolean rank-k in which all the k rank-1 components (rectangles) overlap at a unique middle entry and then replace this entry with a 0 to obtain $\mathbf{X}^{(k)}$. Now $\mathbf{X}^{(k)}$ and $\mathbf{Z}^{(k)}$ are exactly at distance 1 in the squared Frobenius norm and hence $\mathbf{Z}^{(k)}$ is a rank-k factorisation of $\mathbf{X}^{(k)}$ with objective value 1 under objective ζ_F. On the other hand, since exactly k rectangles cover the entry at which $\mathbf{X}^{(k)}$ and $\mathbf{Z}^{(k)}$ differ, if $\mathbf{Z}^{(k)}$ is taken as a rank-k factorisation of $\mathbf{X}^{(k)}$ under objective $\zeta_{(1)}$ it incurs an error of size k. Figure 9.1 shows the idea how to build such a $\mathbf{X}^{(k)}$ for $k = 2, 4, 6$. Each colour corresponds to a rank-1 component and white areas correspond to 0s.



(a) $k = 2$ (b) $k = 4$ (c) $k = 6$

Figure 9.1: Example matrices for which $\zeta_{(1)} = k \cdot \zeta_F$

We first consider the case when k is even. For $k = 2$ take the symmetric matrix $\mathbf{X}^{(2)}$ as in Equation (9.4.2) which corresponds to Figure 9.1a. Since $\mathbf{X}^{(2)}$ has repeated rows and columns, according to Lemma 9.4.1 we may simplify the problem by replacing $\mathbf{X}^{(2)}$ by $\mathbf{X}'^{(2)}$ and recording a weight vector for the rows and columns which indicate how many times each row and column is repeated. This weight vector is then used to update each entry in the objective function with the corresponding weight. For $\mathbf{X}^{(2)}$ the row and column weight vectors coincide as $\mathbf{X}^{(2)}$ is symmetric and we denote it by $\boldsymbol{w}^{(2)}$,

$$\mathbf{X}^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & 1 \end{bmatrix} \Rightarrow \mathbf{X}'^{(2)} = \begin{bmatrix} 1 & 1 & \\ 1 & & 1 \\ & 1 & 1 \end{bmatrix} \text{ with } \boldsymbol{w}^{(2)} = \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix} \qquad (9.4.2)$$

The Boolean rank of $\mathbf{X}^{(2)}$ is 3 and it's isolation number is 3 as the shadowed entries show. Let $\mathbf{Z}^{(2)}$ be obtained from $\mathbf{X}^{(2)}$ by replacing the 0 at entry $(4,4)$ by a 1. $\mathbf{Z}^{(2)}$ clearly has Boolean rank 2, hence it is a feasible rank-2 factorisation of $\mathbf{X}^{(2)}$. Under objective ζ_F $\mathbf{Z}^{(2)}$ incurs an error of size 1, which is optimal as $\zeta_F(\mathbf{X}^{(2)}, 2) \geq 1$ by $\mathbf{X}^{(2)}$ being of Boolean rank-3. On the other hand, under objective $\zeta_{(1)}$, $\mathbf{Z}^{(2)}$ has objective value 2 as the middle entry is covered twice. To see that $\mathbf{Z}^{(2)}$ is optimal under $\zeta_{(1)}$ observe that every entry in $\mathbf{X}'^{(2)}$ apart from the middle entry has weight strictly greater than 2. Hence not covering a 1 of $\mathbf{X}'^{(2)}$ or covering a 0 different from the middle entry incurs an error strictly greater than 2.

For $k > 2$ even let us give a recipe to construct a symmetric matrix $\mathbf{X}'^{(k)}$ and corresponding weight vector $\boldsymbol{w}^{(k)}$. Let $t = \frac{k}{2} - 1$ and let the following $(4t+3) \times (4t+3)$ matrix be $\mathbf{X}'^{(k)}$, where $\mathbf{I}_t$ is the identity matrix of size $t \times t$, $\tilde{\mathbf{I}}_t$ is the reverted identity matrix of size $t \times t$ (so $\tilde{\mathbf{I}}_2 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$) and $\mathbf{J}_t$ is the all 1s matrix of size $t \times t$,

$$\mathbf{X}'^{(k)} = \begin{bmatrix} \mathbf{I}_t & & & \mathbf{1}_t & \tilde{\mathbf{I}}_t & & \\ & 1 & \mathbf{1}_t^\top & 1 & & & \\ & \mathbf{1}_t & \mathbf{J}_t & \mathbf{1}_t & & & \tilde{\mathbf{I}}_t \\ \mathbf{1}_t^\top & 1 & \mathbf{1}_t^\top & 0 & \mathbf{1}_t^\top & 1 & \mathbf{1}_t^\top \\ \tilde{\mathbf{I}}_t & & & \mathbf{1}_t & \mathbf{J}_t & \mathbf{1}_t & \\ & & & 1 & \mathbf{1}_t^\top & 1 & \\ & & \tilde{\mathbf{I}}_t & \mathbf{1}_t & & & \mathbf{I}_t \end{bmatrix}, \qquad \boldsymbol{w}^{(k)} = \begin{bmatrix} (k+1)\mathbf{1}_t \\ (k+1) \\ (k+1)\,\mathbf{1}_t \\ 1 \\ (k+1)\,\mathbf{1}_t \\ (k+1) \\ (k+1)\mathbf{1}_t \end{bmatrix}.$$

$\mathbf{X}'^{(k)}$ has isolation number $\mathfrak{i}(\mathbf{X}'^{(k)}) \geq 2t + 3 = k + 1$ (indicated by the shadowed entries), so no rank-k factorisation can have zero error. Let $\mathbf{Z}'^{(k)}$ be obtained from $\mathbf{X}'^{(k)}$ by replacing the middle 0 by a 1 and let its weight vector be the same as of $\mathbf{X}'^{(k)}$. The Boolean rank of $\mathbf{Z}'^{(k)}$ is then at most k as $\mathbf{Z}'^{(k)} = \mathbf{A}'^{(k)} \circ (\mathbf{A}'^{(k)})^\top$ is an exact factorisation and $\mathbf{A}'^{(k)}$ is of dimension $(4t + 3) \times k$ given by

$$\mathbf{A}'^{(k)} = \begin{bmatrix} \mathbf{I}_t & & & \\ & 1 & & \\ & \mathbf{1}_t & & \tilde{\mathbf{I}}_t \\ \mathbf{1}_t^\top & 1 & 1 & \mathbf{1}_t^\top \\ \tilde{\mathbf{I}}_t & & \mathbf{1}_t & \\ & & 1 & \\ & & & \mathbf{I}_t \end{bmatrix}.$$

This factorisation is illustrated in Figure 9.1 for $k = 4, 6$. Therefore $\mathbf{Z}'^{(k)}$ is a feasible rank-k factorisation of $\mathbf{X}'^{(k)}$. Now $\mathbf{Z}'^{(k)}$ under objective function ζ_F has error 1 and hence it is optimal. In contrast, $\mathbf{Z}'^{(k)}$ evaluated under objective $\zeta_{(1)}$ has error k as the middle 0 is covered k times and it has weight 1. To see that $\mathbf{Z}'^{(k)}$ is optimal under

$\zeta_{(1)}$ as well, note that all entries of $\mathbf{X}'^{(k)}$ apart from the middle 0 have weight strictly greater than k. Therefore, any other rank-k factorisation which does not cover a 1 or covers a 0 which is not the middle 0, incurs an error strictly greater than k, and hence $\mathbf{Z}'^{(k)}$ is optimal under objective $\zeta_{(1)}$ with value $k \cdot \zeta_F$.

For $k = 1$, all 1-BMFs satisfy $\zeta_F(\mathbf{X}, 1) = \zeta_{(1)}(\mathbf{X}, 1)$ by definition. For $k > 1$ odd, we can obtain $\mathbf{X}'^{(k)}$ and $\boldsymbol{w}^{(k)}$ from $\mathbf{X}'^{(k+1)}$ and $\boldsymbol{w}^{(k+1)}$ by removing the first row and column of $\mathbf{X}'^{(k+1)}$ and the corresponding first entry of $\boldsymbol{w}^{(k+1)}$. For $\mathbf{X}'^{(k)}$ then, the same reasoning holds as for k even. □

While Proposition 9.4.2 shows that $\zeta_{(1)}(\mathbf{X}, k)$ can be k times larger than the Frobenius norm objective $\zeta_F(\mathbf{X}, k)$, the matrices in the proof are quite artificial, and in practice we observe that not many zeros are covered by more than a few rank-1 matrices. In fact, our main motivation to consider this new objective function is that we observed that Exponential Formulation I. becomes computationally easier when using objective $\zeta_{(\rho)}$ without compromising the accuracy of factorisations in practice. These numerical observations will be demonstrated in Sections 10.2.2.1 and 10.2.2.2. Therefore let us consider the previously introduced formulations for k-BMF under the new objective $\zeta_{(\rho)}(\mathbf{X}, k)$.

Let us denote a modification of formulation $\mathrm{MIP_F}$ with the new objective function $\zeta_{(\rho)}$ as $\mathrm{MIP}(\rho)$ and use the transformation $\xi_{i,j} = 1 - z_{i,j}$ for $(i, j) \in \mathrm{supp}_1(\mathbf{X})$ to get

$$\begin{aligned}
\zeta_{\mathrm{MIP}(\rho)}(\mathbf{X}, k) = \min_{\xi, q} \quad & \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} \xi_{i,j} + \rho \sum_{(i,j)\in \mathrm{supp}_0(\mathbf{X})} \sum_{r \in \mathcal{R}^{m,n}_{(i,j)}} q_r && (9.4.3)\\
\text{s.t.} \quad & \sum_{r\in\mathcal{R}^{m,n}_{(i,j)}} q_r + \xi_{i,j} \geq 1 \qquad (i,j) \in \mathrm{supp}_1(\mathbf{X}), && (9.4.4)\\
& \sum_{r\in\mathcal{R}^{m,n}} q_r \leq k && (9.4.5)\\
& \xi_{i,j} \geq 0,\ q_r \in \{0,1\} \qquad (i,j) \in \mathrm{supp}_1(\mathbf{X}), r \in \mathcal{R}^{m,n}.
\end{aligned}$$

One of the imminent advantages of using objective $\zeta_{(\rho)}$ is that we need only declare variables for entries $(i, j) \in \mathrm{supp}_1(\mathbf{X})$ and can consequently delete the weak constraints (9.2.3) from the formulation. The LP relaxation of $\mathrm{MIP}(\rho)$ ($\mathrm{MLP}(\rho)$) is obtained by giving up on the integrality constraints on q_r and observing that without loss of generality we can simply write $q_r \geq 0$ for all $r \in \mathcal{R}^{m,n}$. We next show that the optimal solutions of the LP relaxation of $\mathrm{MIP_F}$ and $\mathrm{MLP}(\rho)$ with $\rho = \frac{1}{k}$ coincide.

Proposition 9.4.3. *The optimal solutions of the LP relaxations* $\mathrm{MLP_F}$ *and* $\mathrm{MLP}(\frac{1}{k})$ *coincide.*

Proof. It suffices to observe that as $\mathrm{MLP_F}$ is a minimisation problem, each $z_{i,j}$ $(i,j) \in \mathrm{supp}_0(\mathbf{X})$ takes the value $\frac{1}{k}\sum_{r\in\mathcal{R}^{m,n}_{(i,j)}} q_r$ in any optimal solution to $\mathrm{MLP_F}$ due to constraint (9.2.3). This implies that the second terms in the objective function (9.2.1) of $\mathrm{MIP_F}$ and (9.4.3) of $\mathrm{MLP}(\frac{1}{k})$ have the same value. □

Therefore one may instead solve $\mathrm{MLP}(\frac{1}{k})$ that has fewer variables and constraints than $\mathrm{MLP_F}$. In addition, for all $\rho > 0$, a corollary of Proposition 9.2.1 holds by looking at the dual of $\mathrm{MLP}(\rho)$ ($\mathrm{MDP}(\rho)$). Let us associate variables $p_{i,j}$ for $(i,j) \in \mathrm{supp}_1(\mathbf{X})$ to constraints (9.4.4) and variable μ to constraint (9.4.5). Then the dual of $\mathrm{MLP}(\rho)$ is ($\mathrm{MDP}(\rho)$):

$$\begin{aligned}
\zeta_{\mathrm{MDP}(\rho)}(\mathbf{X},k) = &\max_{p,\mu} \sum_{(i,j)\in\mathrm{supp}_1(\mathbf{X})} p_{i,j} - k\,\mu \\
&\text{s.t.} \sum_{\substack{(i,j)\in\\ \mathrm{supp}_1(\mathbf{X})\cap\mathrm{supp}_1(\mathbf{R})}} p_{i,j} - \mu \le \rho\cdot|\,\mathrm{supp}_0(\mathbf{X})\cap\mathrm{supp}_1(\mathbf{R})| \quad \mathbf{R}\in\mathcal{R}^{m,n}, \qquad (9.4.6)\\
&\qquad \mu \ge 0,\ p_{i,j}\in[0,1] \qquad\qquad (i,j)\in\mathrm{supp}_1(\mathbf{X}).
\end{aligned}$$

Corollary 9.4.4. *Let* $\mathbf{X} \in \{0,1,?\}^{m\times n}$ *and* $k \in \mathbb{Z}_{++}$*. If* $\mathfrak{i}(\mathbf{X}) > k$*, then for all* $\rho > 0$ *the optimal objective value of the LP relaxation of* $\mathrm{MIP}(\rho)$ *satisfies*

$$\zeta_{\mathrm{MLP}(\rho)}(\mathbf{X},k) \ge \rho\cdot(\mathfrak{i}(\mathbf{X}) - k)\,.$$

Proof. The proof is a simple modification of Proposition 9.2.1's proof. The dual of $\mathrm{MLP}(\rho)$ ($\mathrm{MDP}(\rho)$) differs from $\mathrm{MDP_F}$ by having the constant value ρ instead of dual variables $s_{i,j}$ and constraints (9.4.6) instead of (9.2.6). Therefore setting $p_{i,j} = \rho$ for all $(i,j) \in S$ and 0 otherwise (where S is a maximum isolated set of $\mathbf{X}$), and $\mu = \rho$ gives the required bound of $\rho\,(\mathfrak{i}(\mathbf{X}) - k)$. □

Chapter 10

Computational approach and experiments

The integer programs introduced in the previous sections provide a framework for computing k-BMF with dual bounds. In this chapter, we present a computational approach to solve Exponential formulation I. despite it having an exponential number of variables. Then we present some experimental results to demonstrate the practical applicability of integer programming to obtain low-error factorisations. More specifically we detail our pricing strategies during the column generation process and present a thorough comparison of models MIP_{F}, $\text{MIP}(\rho)$ and CIP on synthetic and real world datasets. Our code and data can be downloaded from [59].

10.1 Column generation

It is clearly not practical to solve the master integer program $\text{MIP}(\rho)$ or its LP relaxation $\text{MLP}(\rho)$ explicitly as the formulation has an exponential number of variables. *Column generation* (CG) is a well-known technique to solve large LPs iteratively by only considering the variables which have the potential to improve the objective function [7]. The column generation procedure is initialised by solving a *Restricted* Master LP (RMLP) which has a small subset of the variables of the full problem. The next step is to identify a missing variable with *negative reduced cost* to be added to RMLP. To avoid considering all missing variables explicitly, a *pricing problem* is formulated and solved. The solution of the pricing problem either returns a variable with negative reduced cost and the procedure is iterated; or proves that no such variable exists and hence the solution of RMLP is optimal for the full MLP. In this section, we detail how CG technique can be used to solve the LP relaxation of $\text{MIP}(\rho)$ iteratively.

Each *Restricted* MLP(ρ) (RMLP(ρ)) has the same number of constraints as the full MLP(ρ) and all variables $\xi_{i,j}$ for $(i,j) \in \mathrm{supp}_1(\mathbf{X})$ but it only has a small subset of variables q_r for $r \in \mathcal{R}' \subset \mathcal{R}^{m,n}$ where $|\mathcal{R}'| \ll |\mathcal{R}^{m,n}|$. Recall that each variable q_r corresponds to a rank-1 binary matrix $r \in \mathcal{R}^{m,n}$ which determines the coefficients of q_r in the constraints as well as the objective function. Hence at every iteration of the CG procedure we either need to find a rank-1 binary matrix for which the associated variable has a negative reduced cost, or, prove that no such matrix exists.

The pricing problem. At the first iteration of CG, RMLP(ρ) may be initialised with $\mathcal{R}' = \emptyset$ or can be warm started by identifying a few rank-1 matrices in $\mathcal{R}^{m,n}$ using a heuristic. After solving the RMLP(ρ) to optimality via a standard LP solver, one obtains an optimal dual solution $[\boldsymbol{p}^*, \mu^*]$ to the current RMLP(ρ). To identify a missing variable q_r that has negative reduced cost, we solve the following pricing problem (PP):

$$\begin{aligned} \text{(PP)}\ \ \omega(\mu^*, \boldsymbol{p}^*) = \mu^* - \max_{a,b,y} & \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} p^*_{ij} y_{i,j} - \rho \sum_{(i,j)\in \mathrm{supp}_0(\mathbf{X})} y_{i,j} & \\ \text{s.t. } & y_{i,j} = a_i b_j, & i \in [n], j \in [m], \\ & a_i, b_j \in \{0,1\}, & i \in [n], j \in [m]. \end{aligned}$$

PP may be formulated as an integer linear program (IP$_{\mathrm{PP}}$) by using McCormick envelopes [82]. The objective of PP depends on the current dual solution $[\boldsymbol{p}^*, \mu^*]$ and its optimal solution corresponds to a rank-1 binary matrix $\boldsymbol{a}\boldsymbol{b}^\top = r \in \mathcal{R}^{m,n}$ whose corresponding variable q_r in MLP(ρ) has the smallest reduced cost. If $\omega(\mu^*, \boldsymbol{p}^*) \geq 0$, then the current RMLP($\rho$) does not have any missing variables with negative reduced cost and consequently the current solution of RMLP(ρ) is optimal for MLP(ρ). If $\omega(\mu^*, \boldsymbol{p}^*) < 0$, then the variable q_r associated with the rank-1 binary matrix $r = \boldsymbol{a}\boldsymbol{b}^\top$ is added to the next RMLP(ρ) and the procedure is iterated. Moreover, any feasible solution to PP with a negative reduced cost can (also) be added to the RMLP(ρ) to continue the procedure. CG terminates with a proof of optimality if at some iteration we have $\omega(\mu^*, \boldsymbol{p}^*) \geq 0$.

We mention that the pricing problem is essentially as hard as CIP$_1$, since it has the same exact formulation except for the different linear objective. This shows that we can expect that solving PP to optimality will be a bottle neck in our column generation approach. On the other hand, via our column generation approach we can compute a rank-k factorisation by reducing it to solving a series of problems that are equivalent to the rank-1 case.

Solving the master integer program. After the CG process, if the optimal solution of MLP(ρ) is integral, then it also is optimal for MIP(ρ). However, if it is fractional, then this solution only provides a lower bound on the optimal value of MIP(ρ). In this case we obtain an integer feasible solution by solving a *Restricted* MIP(ρ) (RMIP(ρ)) over the rank-1 binary matrices generated by the CG process applied to MLP(ρ). This integer feasible solution is optimal for MIP(ρ) provided that the objective value of RMIP(ρ) is equal to the ceiling of the objective value of MLP(ρ). If this is not the case, one needs to embed CG into a branch-and-bound tree [80] to solve MIP(ρ) to optimality, which is a relatively complicated process and we do not consider it in this thesis.

Computing lower bounds. Note that even if the CG procedure is terminated prematurely, one can still obtain a lower bound on MLP(ρ) and therefore on MIP(ρ) by considering the dual of MLP(ρ). Let the objective value of of the current RMLP(ρ) be

$$\zeta_{\mathrm{RMLP}(\rho)}(\mathbf{X},k) = \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} \xi^*_{i,j} + \rho \sum_{(i,j)\in \mathrm{supp}_0(\mathbf{X})} \sum_{r\in \mathcal{R}^{m,n}_{(i,j)}} q^*_r = \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} p^*_{i,j} - k\cdot \mu^*$$

where $[\boldsymbol{\xi}^*, \boldsymbol{q}^*]$ is the optimal solution of RMLP(ρ) and $[\boldsymbol{p}^*, \mu^*]$ is the corresponding optimal dual solution which does not necessarily satisfy all of the constraints (9.4.6) for MDP(ρ). Now assume that we solve PP to optimality and obtain a rank-1 binary matrix with a negative reduced cost, $\omega(\mu^*, \boldsymbol{p}^*) < 0$. In this case, we can construct a feasible solution $[\boldsymbol{p}, \mu]$ to MDP(ρ) by setting $\boldsymbol{p} := \boldsymbol{p}^*$ and $\mu := \mu^* - \omega(\mu^*, \boldsymbol{p}^*)$ and obtain the following bound on the optimal value $\zeta_{\mathrm{MLP}(\rho)}(\mathbf{X},k)$ of MLP(ρ),

$$\begin{aligned} \zeta_{\mathrm{MLP}(\rho)}(\mathbf{X},k) &\geq \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} p_{i,j} - k\,\mu \\ &= \sum_{(i,j)\in \mathrm{supp}_1(\mathbf{X})} p^*_{i,j} - k\,\left(\mu^* - \omega(\mu^*, \boldsymbol{p}^*)\right) \\ &= \zeta_{\mathrm{RMLP}(\rho)}(\mathbf{X},k) + k\,\omega(\mu^*, \boldsymbol{p}^*). \end{aligned} \tag{10.1.1}$$

If we do not have the optimal solution to PP but have a lower bound $\underline{\omega}(\mu^*, \boldsymbol{p}^*)$ on it, $\omega(\mu^*, \boldsymbol{p}^*)$ can be replaced by $\underline{\omega}(\mu^*, \boldsymbol{p}^*)$ in Equation (10.1.1) and the bound on MLP(ρ) still holds. Furthermore, this lower bound on MLP(ρ) naturally provides a valid lower bound on MIP(ρ), thus giving us a bound on the optimality gap.

Column generation for $\mathrm{MLP}_{\mathbf{F}}$ The CG approach is described above as applied to the LP relaxation of MIP(ρ). To apply CG to $\mathrm{MLP}_{\mathrm{F}}$ only a small modification needs to be done. The Restricted $\mathrm{MLP}_{\mathrm{F}}$ provides dual variables for constraints (9.2.3) which are used in the objective of PP for coefficients of $y_{i,j}$ for $(i,j) \in \mathrm{supp}_0(\mathbf{X})$.

We note that CG cannot be used to solve the LP relaxation of the *strong formulation* of $\mathrm{MIP}_{\mathrm{F}}$ in which constraints (9.2.3) are replaced by exponentially many constraints $q_r \leq z_{i,j}$ for all $r \in \mathcal{R}^{m,n}_{(i,j)}$ and $(i,j) \in \mathrm{supp}_0(\mathbf{X})$. This is due to the fact that CG could cycle and generate the same column over and over again. For example, consider applying CG to solve the strong formulation of $\mathrm{MLP}_{\mathrm{F}}$ and start with the rank-1 binary matrix of all 1s as the first column associated with variable q_1. The objective value of the corresponding Restricted $\mathrm{MLP}_{\mathrm{F}}$ would be $\zeta^{(1)}_{\mathrm{RMLP}}(\mathbf{X},k) = 0 + |\,\mathrm{supp}_0(\mathbf{X})|$ for the solution vector $[\boldsymbol{\xi}^{(1)}, \boldsymbol{z}^{(1)}, \boldsymbol{q}^{(1)}] = [\mathbf{0}, \mathbf{1}, 1]$ as all entries of the input matrix are covered. Adding the same rank-1 binary matrix of all 1s in the next iteration and setting $[q_1, q_2] = [\frac{1}{2}, \frac{1}{2}]$, allows us to keep $\boldsymbol{\xi}^{(2)} = \mathbf{0}$ but reduce the value of $\boldsymbol{z}^{(2)}$ to $\frac{1}{2}\mathbf{1}$ to obtain an objective value $\zeta^{(2)}_{\mathrm{RMLP}}(\mathbf{X},k) = 0 + \frac{1}{2}|\,\mathrm{supp}_0(\mathbf{X})|$. Therefore, repeatedly adding the same matrix of all 1s for t iterations, the objective function would become $\zeta^{(t)}_{\mathrm{RMLP}}(\mathbf{X},k) = 0 + \frac{1}{t}|\,\mathrm{supp}_0(\mathbf{X})|$ for the solution vector $[\boldsymbol{\xi}^{(t)}, \boldsymbol{z}^{(t)}, \boldsymbol{q}^{(t)}] = [\mathbf{0}, \frac{1}{t}\mathbf{1}, \frac{1}{t}\mathbf{1}]$. Consequently, as $t \to \infty$ we would have $\zeta^{(t)}_{\mathrm{RMLP}}(\mathbf{X},k) \to 0$ and during the column generation process we repeatedly generate the same rank-1 binary matrix.

Heuristics for the pricing problem Generating rank-1 binary matrices with negative reduced cost efficiently is at the heart of the CG process. The pricing problem for MLP(ρ) can be formulated as a Bipartite Binary Quadratic Program (BBQP),

$$(\mathrm{QP}_{\mathrm{PP}}) \quad \omega(\mu^*, \boldsymbol{p}^*) = \mu^* - \max_{\boldsymbol{a}\in\{0,1\}^m, \boldsymbol{b}\in\{0,1\}^n} \boldsymbol{a}^\top \boldsymbol{\mathcal{H}} \boldsymbol{b}. \tag{10.1.2}$$

with $\mathcal{H}_{i,j} = p^*_{i,j} \in [0,1]$ for $(i,j) \in \mathrm{supp}_1(\mathbf{X})$, $\mathcal{H}_{i,j} = -\rho$ for $(i,j) \in \mathrm{supp}_0(\mathbf{X})$ and $\mathcal{H}_{i,j} = 0$ for $(i,j) \notin \Omega(\mathbf{X})$. Since any rank-1 binary matrix with negative reduced cost is valid to be added as a column to the next RMLP, we may use Algorithm 3 and its modifications mentioned in Section 8.3.2 to provide a warm start to PP at every iteration of CG.

A heuristic for k-BMF. In addition, in some cases it is useful to add a few columns for MLP(ρ) before starting the CG process so that set $\mathcal{R}'$ is not completely empty. In Algorithm 4, we give a new heuristic for k-BMF which sequentially finds k rank-1 binary matrices using the greedy algorithm for BBQP as a subroutine. We refer to this heuristic as the k-*Greedy* method.

Algorithm 4: Greedy algorithm for k-BMF (k-Greedy)

```
Input: $\mathbf{X} \in \{0, 1, ?\}^{m \times n}$, $k \in \mathbb{Z}_{++}$.
Set $\boldsymbol{\mathcal{W}} \in \{-1, 0, 1\}^{m \times n}$ to $\mathcal{W}_{i,j} = 2x_{i,j} - 1$ for $(i, j) \in \Omega(\mathbf{X})$ and $\mathcal{W}_{i,j} = 0$
otherwise.
for $\ell \in [k]$ do
    $\boldsymbol{a}, \boldsymbol{b} = \mathrm{BBQP}(\boldsymbol{\mathcal{W}})$ // compute a 1-BMF via Algorithm 3
    $\mathbf{A}_{:,\ell} = \boldsymbol{a}$
    $\mathbf{B}_{\ell,:} = \boldsymbol{b}^\top$
    $\boldsymbol{\mathcal{W}}[\boldsymbol{a}\boldsymbol{b}^\top == 1] = 0$ // set entries of $\boldsymbol{\mathcal{W}}$ to zero that are covered
end
Output: $\mathbf{A} \in \{0, 1\}^{m \times k}$, $\mathbf{B} \in \{0, 1\}^{k \times n}$
```

10.2 Experiments

10.2.1 Data

If $\mathbf{X}$ contains rows (or columns) of all zeros, deleting these rows (or columns) leads to an equivalent problem whose solution $\mathbf{A}$ and $\mathbf{B}$ can easily be translated to a solution for the original problem by inserting a row of zeros to $\mathbf{A}$ (respectively a column of zeros to $\mathbf{B}$) in the corresponding place. In addition, if $\mathbf{X}$ contains duplicate rows or columns, by Lemma 9.4.1 there is an optimal rank-k factorisation which has the same row-column repetition pattern as $\mathbf{X}$. Hence we solve the problem on a smaller matrix $\mathbf{X}'$ which is obtained from $\mathbf{X}$ by keeping only one copy of each row and column, and use an updated objective function in which every entry is weighted proportional to the number of rows and columns it is contained in $\mathbf{X}$.

Synthetic data. We build our dataset of binary matrices with prescribed sparsity and Boolean rank as follows. To get a matrix $\mathbf{X} \in \{0, 1\}^{m \times n}$ with Boolean rank at most κ, first we randomly generate two binary matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ of dimension $m \times \kappa$ and $\kappa \times n$, then compute their Boolean product to get $\mathbf{X}$. This ensures $\mathbf{X}$ has Boolean rank at most κ. To obtain a certain sparsity for $\mathbf{X}$, we control the probability of entries of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ being zero. More specifically, if we generate $\tilde{a}_{i,\ell}$, $\tilde{b}_{\ell,j}$ to be zero with probability p, then $x_{i,j} = \bigvee_{\ell=1}^{\kappa} \tilde{a}_{i,\ell}\tilde{b}_{\ell,j}$ is zero with probability $(1 - (1 - p)^2)^\kappa$. Hence, to obtain $\mathbf{X}$ with σ percent of zeros, we need to generate entries of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ to be zero with probability $p = 1 - \sqrt{1 - (\sigma/100)^{\frac{1}{\kappa}}}$.

We generate matrices as described above with $n = 20$ columns and $\kappa = 10$. The number of rows (m) is set to be $20, 35$ or 50. For each of the three dimensions $(20 \times 20, 35 \times 20, 50 \times 20)$, we generate 10 *sparse* matrices with 75% zeroes and

10 *normal* matrices with 50% zeroes, corresponding to 10 different seed settings in the random number generation. We call this initial set of $2 \cdot 3 \cdot 10$ matrices the *clean* matrices. Next, we create a set of *noisy* matrices from the clean matrices by randomly flipping 5% of the entries of each matrix. The noisy matrices are not necessarily of Boolean rank at most $\kappa = 10$, but they are at most $0.05 \cdot m \cdot n$ squared Frobenius distance away from a Boolean rank 10 matrix. Therefore, our test bed consists of 120 matrices corresponding to 2 noise level settings (*noisy* or *clean*), 2 sparsity levels (*sparse* or *normal*), 3 dimensions $(20 \times 20, 35 \times 20, 50 \times 20)$ and 10 random seeds. Applying the preprocessing steps to our synthetic dataset achieves the largest dimension reduction on clean matrices, while the dimension of noisy matrices scarcely changes. A table summarising the parameters used to generate our data can be found in Appendix B.1.

Real world data. We work with eight real world categorical datasets that were downloaded from online repositories [31, 64]. In general if a dataset has a categorical feature C with N discrete options v_j, $(j \in [N])$, we convert feature C into N binary features B_j $(j \in N)$ so that if the i-th sample takes option v_j for C that is $(C)_i = v_j$, then we have $(B_j)_i = 1$ and $(B_\ell)_i = 0$ for all $\ell \neq j \in [N]$. This technique of binarisation of categorical columns has been applied in [61] and [5]. If a row i has a missing value in the column of feature C, we leave the corresponding binary feature columns with missing values in row i. Table 10.1 shows a short summary of the resulting full-binary datasets used, in-depth details on converting categorical columns into binary, missing value treatment and feature descriptions can be found in Appendix B.2.

	zoo	tumor	hepatitis	heart	lymp	audio	apb	votes
$m \times n$	101 × 17	339 × 24	155× 38	242× 22	148×44	226 × 92	105× 105	435 ×16
# missing	0	670	334	0	0	899	0	392
%1s	44.3	24.3	47.2	34.4	29.0	11.3	8.0	49.2

Table 10.1: Summary of binary real world datasets

10.2.2 Testing the computational approach to exponential formulation I.

Since the efficiency of CG greatly depends on the speed of generating columns, let us illustrate the speed-up gained by using heuristics to solve the pricing problem. At each iteration of CG procedure, 30 variants of Algorithm 3 are computed to obtain an initial feasible solution to the pricing problem. The 30 variants of the greedy algorithm use the original and revised ordering, their transpose and perturbed version and 22 random orderings. All greedy solutions are improved by the alternating heuristic until no further improvement is found.

Under *exact* pricing, the best heuristic solution is used as a warm start and $\mathrm{IP}_{\mathrm{PP}}$ is solved to optimality at each iteration using CPLEX [23]. In simple heuristic (*heur*) pricing, if the best heuristic solution to PP has negative reduced cost then it is directly added to the next RMLP(ρ). If at some iteration, the best heuristic column does not have negative reduced cost, CPLEX is used to solve $\mathrm{IP}_{\mathrm{PP}}$ to optimality for that iteration. The multiple heuristic (*heur_multi*) pricing strategy is a slight modification of the simple heuristic strategy, in which at each iteration all columns with negative reduced cost are added to the next RMLP(ρ).

Figure 10.1 indicates the differences between pricing strategies when solving MLP(1) via CG for $k = 5, 10$ on the zoo dataset. The primal objective value of MLP(1) (decreasing curve) and the value of the dual bound (increasing curve) computed using the formula in Equation (10.1.1) are plotted against time. Sharp increases in the dual bound for heuristic pricing strategies correspond to iterations in which CPLEX was used to solve $\mathrm{IP}_{\mathrm{PP}}$, as for the evaluation of the dual bound on MLP(1) a lower bound on $\omega(\mu^*, \boldsymbol{p}^*)$ is needed which heuristic solutions do not provide. While we observe a tailing off effect [80] on all three curves, both heuristic pricing strategies provide a significant speed-up from exact pricing, adding multiple columns at each iteration being the fastest.

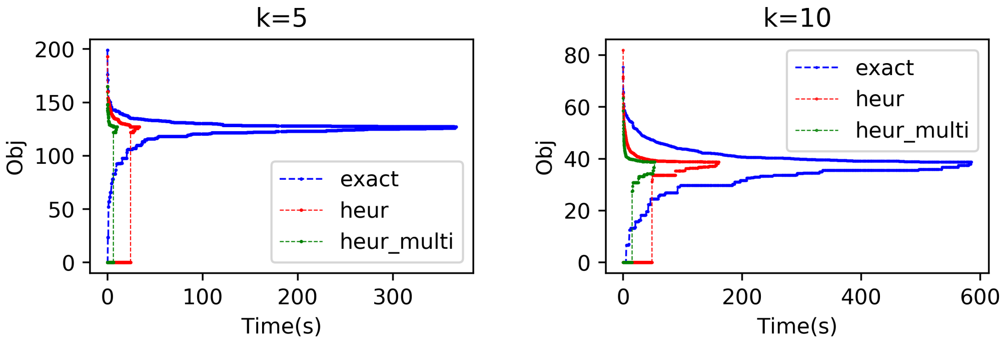


Figure 10.1: Comparison of pricing strategies for solving MLP(1) on the zoo dataset

In order for CG to terminate with a certificate of optimality, at least one pricing problem has to be solved to optimality. Unfortunately for larger datasets we cannot expect this to be achieved in a short amount of time. Therefore, we change the multiple heuristic pricing strategy to get a pricing strategy that we use in the rest of the experiments as follows. We impose an overall fixed time limit on the CG process and use the barrier method in CPLEX as the LP solver for RMLP at each iteration. At each iteration of CG, we add up to 2 columns with the most negative reduced cost to the next RMLP. If at an iteration, heuristics for PP do not provide a column with negative reduced cost and CPLEX has to be used to improve the heuristic solution, we do not solve $\mathrm{IP}_{\mathrm{PP}}$ to optimality but abort CPLEX after 25 seconds if a column with negative reduced cost has been found. While these modifications result in a speed-up, they reduce the chance of obtaining a strong dual bound. In case we wish to focus more on computing a stronger dual bound on MLP, we may continue solving $\mathrm{IP}_{\mathrm{PP}}$ via CPLEX even when a heuristic negative reduced cost solution is available.

10.2.2.1 MLP(1) vs $\mathbf{MLP_F}$

In this section we compare the LP relaxations of MIP(1) and $\mathrm{MIP}_{\mathrm{F}}$. According to Proposition 9.4.3 the optimal solution of $\mathrm{MLP}_{\mathrm{F}}$ is equivalent to $\mathrm{MLP}(\frac{1}{k})$ and hence we solve $\mathrm{MLP}(\frac{1}{k})$ which has fewer variables and constraints than $\mathrm{MLP}_{\mathrm{F}}$. To solve MLP(1) and $\mathrm{MLP}(\frac{1}{k})$, we start off from 0 rank-1 binary matrices so $\mathcal{R}' = \emptyset$ in the first RMLP and set a total time limit of 600 seconds, so we either solve MLP to optimality under 600 seconds or run out of time and compute the gap between the last RMLP and the best dual bound MDP according to formula

$$100 \cdot \frac{\zeta_{\mathrm{RMLP}}(\mathbf{X}, k) - \zeta_{\mathrm{MDP}}(\mathbf{X}, k)}{\zeta_{\mathrm{RMLP}}(\mathbf{X}, k)}.$$

As MLP(1) and $\mathrm{MLP}(\frac{1}{k})$ correspond to the LP relaxations of MIP(1) and $\mathrm{MIP}_{\mathrm{F}}$ with integral objective coefficients, any fractional dual bound may be rounded up to give a valid bound on the master IP. Therefore, we stop CG whenever the ceiling of the dual bound reaches the objective value of RMLP.

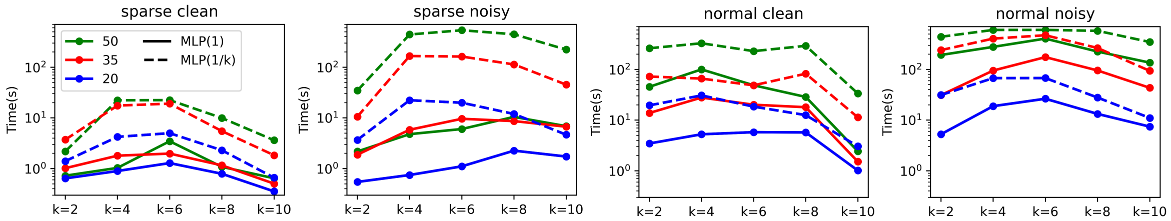


Figure 10.2: Time taken in seconds to solve MLP(1) and $\mathrm{MLP}(\frac{1}{k})$ via CG on synthetic data

Figure 10.2 shows the time taken in seconds on a logarithmic scale to solve MLP(1) and MLP($\frac{1}{k}$) via CG for $k = 2, 4, \ldots, 10$ on the synthetic matrices. Each line corresponds to the average taken over 10 instances with the same dimension, sparsity and noise level. Blue lines correspond to matrices of dimension 20×20, red to 35×20 and green to 50×20. Solid lines are used for MLP(1) and dashed for MLP($\frac{1}{k}$). First, we observe that it is significantly faster to solve both MLPs on *sparse* and *clean* matrices as opposed to *normal* and *noisy* ones of the same dimension. Preprocessing is more effective in reducing the dimension for clean matrices in comparison to noisy ones (see Table B.1 in Appendix B.1) which explains why noisy instances take longer. In addition, both MLP(1) and MLP($\frac{1}{k}$) have a number of variables and constraints directly proportional to non-zero entries of the input matrix, hence a sparse input matrix requires a smaller problem to be solved. Second, we see that $k = 10$ are solved somewhat faster. This can be explained by all matrices in our test bed being generated to have Boolean rank at most 10. For a rank-10 factorisation of *clean* matrices without noise we get 0 factorisation error under both models MIP(1) and MIP$_F$ and hence LP relaxation objective value 0. For *noisy* matrices we observe the error to be in line with our expectation of $0.05 \cdot m \cdot n$. We observe that in some cases it takes significantly longer to solve MLP($\frac{1}{k}$), and in all ten instances of 50×20 *normal-noisy* matrices MLP($\frac{1}{k}$) for $k = 6$ runs out of the time budget of 600 sec. In the experiments, we see the amount of time CG takes is directly proportional to the number of columns generated, MLP($\frac{1}{k}$) generating significantly more columns than MLP(1).

10.2.2.2 Obtaining integral solutions

Once we obtain some rank-1 binary matrices (i.e. columns) via CG applied to a master LP, we can obtain an integer feasible solution by solving either of the master IPs over the columns available. Here we explore obtaining integer feasible solutions by solving MIP(1) and MIP$_F$ over the columns generated by formulations MLP(1) and MLP($\frac{1}{k}$). We use CPLEX as our integer program solver and set a total time limit of 300 seconds.

Figure 10.3 shows the factorisation error in $\| \cdot \|_F^2$ of integer feasible solutions obtained by solving MIP(1) over columns generated by MLP(1) and MLP($\frac{1}{k}$). As previously, each line corresponds to the average taken over 10 matrices with same dimension, sparsity and noise level. Solid lines are used to denote where the columns used were generated by MLP(1) and dashed where by MLP($\frac{1}{k}$). Comparing the error values of the dashed and solid lines we draw a crucial observation: columns generated

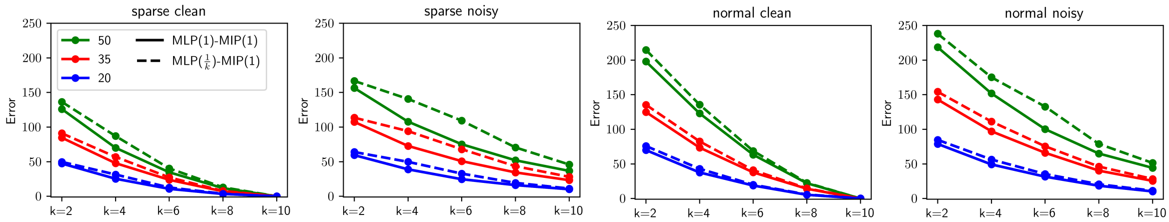


Figure 10.3: Factorisation error in $\|\cdot\|_F^2$ of integral solutions by MIP(1) from columns by MLP(1) and MLP($\frac{1}{k}$)

by MLP(1) seem to be a better basis for obtaining low-error integer feasible solutions than columns by MLP($\frac{1}{k}$). We suspect this is the case as in the majority of rank-k factorisations most entries are only covered by a few rank-1 binary matrices whereas MLP($\frac{1}{k}$) favours rank-1 matrices which heavily cover 0 entries of the input matrix. This is because the coefficient in MLP($\frac{1}{k}$)'s objective function corresponding to a zero entry at position (i, j) is only $\frac{1}{k}\times$(number of rank-1 matrices covering (i, j)), hence it is cheaper for MLP($\frac{1}{k}$) to cover a 0 by a few (less than k) rank-1 matrices than to leave any 1s uncovered. We also conducted a set of experiments using formulation $\mathrm{MIP_F}$ and we see that the factorisation error when using formulation MIP(1) to obtain the integral solutions is extremely close to that of $\mathrm{MIP_F}$, see Appendix B.3 Tables B.2 and B.3 for the precise difference in the factorisation error between the two master IPs.

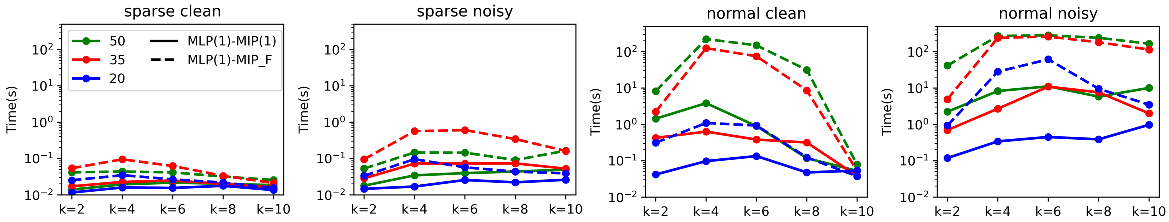


Figure 10.4: Time taken in seconds to solve MIP(1) and MIP_F on columns generated by MLP(1)

Figure 10.4 shows the time taken to solve the master IPs on columns generated by MLP(1). We observe that MIP(1) takes notably faster to solve than MIP_F and on most normal-noisy matrices MIP_F runs out of the time budget of 300 seconds. Solving both master IPs on columns by MLP($\frac{1}{k}$) also shows us that while solving MIP(1) over a larger set of columns adds only a few seconds for most instances, MIP_F runs out of the time budget of 300 secs in about half the cases, see Appendix Table B.3. These observations suggest using MIP(1) to find integer feasible solutions in the future as the solution quality is extremely close to that of MIP_F but at a fraction of computational effort.

10.2.3 Accuracy and speed of the IP formulations

In this section we computationally compare the integer programs introduced in Section 9. CIP due to its polynomial size can be directly given to a general purpose IP solver like CPLEX and we set a time limit of 600 seconds on its running time. We expect solution times for CIP to grow proportional to k and density of $\mathbf{X}$ according to Proposition 9.1.1. Similarly, we may try to attack the exponential formulation EIP directly by CPLEX. Since however EIP requires the complete enumeration of 2^n binary vectors for an input matrix $\mathbf{X}$ of size $m \times n$ we can only solve its root LP under 600 seconds in a very few cases. For these few cases however, we observe the objective value of ELP to agree with $\text{MLP}(\frac{1}{k})$, which gives an experimental confirmation of Proposition 9.3.1. In the following experiments, formulation MIP_F is used on columns generated by $\text{MLP}(\frac{1}{k})$, while MIP(1) on columns by MLP(1). The final solution of MIP(1) is evaluated under the original $\|\cdot\|_F^2$ objective and that error is reported. As previously, the master LPs are solved with a time limit of 600 seconds and the master IPs with an additional time limit of 300 seconds.

data		k=2			k=5			k=10	
(n-sparsity-noise)	MIP_F	MIP(1)	CIP	MIP_F	MIP(1)	CIP	MIP_F	MIP(1)	CIP
20-sparse-clean	49.6	**47.4**	**47.4**	20.8	**16.6**	16.7	**0.0**	**0.0**	**0.0**
20-sparse-noisy	64.0	59.5	**59.3**	42.6	**30.3**	30.7	11.2	**10.2**	10.3
20-normal-clean	75.0	70.0	**68.7**	30.6	27.7	**26.5**	0.3	0.3	**0.0**
20-normal-noisy	84.6	78.9	**77.2**	47.3	40.2	**40.1**	11.2	**10.7**	11.2
35-sparse-clean	90.9	**84.7**	**84.7**	39.1	**34.5**	34.9	0.1	**0.0**	**0.0**
35-sparse-noisy	113.4	107.5	**106.9**	84.4	**60.5**	61.7	28.4	**23.3**	27.1
35-normal-clean	134.2	125.0	**121.7**	64.5	54.1	**53.4**	**0.0**	**0.0**	**0.0**
35-normal-noisy	153.6	143.1	**139.1**	101.7	**80.3**	81.7	31.1	**25.5**	31.1
50-sparse-clean	136.0	126.1	**125.6**	61.4	**50.6**	51.5	0.1	**0.0**	**0.0**
50-sparse-noisy	166.2	**156.5**	156.7	135.0	**89.8**	93.9	49.6	**36.7**	41.4
50-normal-clean	215.1	198.0	**194.3**	106.1	**91.0**	95.0	**0.0**	**0.0**	**0.0**
50-normal-noisy	237.2	218.6	**214.2**	168.6	123.9	**123.4**	62.2	**44.3**	61.3

Table 10.2: Factorisation error in $\|\cdot\|_F^2$ of solutions obtained via formulations MIP_F, MIP(1) and CIP_F

Table 10.2 shows the factorisations error in $\|\cdot\|_F^2$ obtained by MIP_F, MIP(1) and CIP and Table 10.3 shows the corresponding solution times in seconds. Each row of Table 10.2 and 10.3 corresponds to the average of 10 synthetic matrices of the same size, sparsity and noise. The lowest error results are indicated in boldface. We

data		k=2			k=5			k=10	
(n-sparsity-noise)	$\mathrm{MIP}_{\mathrm{F}}$	MIP(1)	CIP	$\mathrm{MIP}_{\mathrm{F}}$	MIP(1)	CIP	$\mathrm{MIP}_{\mathrm{F}}$	MIP(1)	CIP
20-sparse-clean	1.1	0.4	1.6	4.6	0.4	169.7	0.7	0.4	1.9
20-sparse-noisy	2.7	0.6	21.8	233.7	0.8	601.6	10.9	1.8	602.9
20-normal-clean	15.2	3.5	56.2	303.2	5.4	600.3	3.3	1.0	15.8
20-normal-noisy	31.3	5.4	295.5	336.6	17.6	600.8	65.2	8.0	602.0
35-sparse-clean	4.0	0.8	17.3	108.4	0.9	449.8	1.9	0.5	5.3
35-sparse-noisy	12.1	1.9	147.8	514.0	6.4	602.3	275.1	6.8	605.2
35-normal-clean	76.0	14.2	188.6	378.5	21.8	600.8	23.2	1.6	80.6
35-normal-noisy	195.3	31.8	589.7	739.3	132.1	600.7	394.7	45.3	602.4
50-sparse-clean	2.6	0.6	21.9	176.3	1.1	519.9	3.8	0.7	12.9
50-sparse-noisy	28.1	2.2	285.4	827.7	6.6	602.3	523.9	6.9	605.1
50-normal-clean	362.0	46.8	509.9	692.1	153.6	602.1	187.2	2.5	139.4
50-normal-noisy	601.6	194.8	578.2	903.9	341.1	601.0	649.8	146.2	601.6

Table 10.3: Time in seconds to obtain solutions in Table 10.2 via formulations $\mathrm{MIP}_{\mathrm{F}}$, MIP(1) and $\mathrm{CIP}_{\mathrm{F}}$

observe that MIP(1) provides the lowest error factorisation in most cases, but CIP gives the lowest error when only looking at $k = 2$. The significantly higher error values of $\mathrm{MIP}_{\mathrm{F}}$ are due to the lower quality columns generated by $\mathrm{MLP}(\frac{1}{k})$ on which it is solved and also partly due to the fact that it is slower to solve $\mathrm{MIP}_{\mathrm{F}}$ than solving MIP(1). We emphasise that we do not do branch-and-price when solving MIP(1) or $\mathrm{MIP}_{\mathrm{F}}$. Table 10.3 shows that MIP(1) is the fastest in all cases, while CIP runs out of its time limit on all noisy instances for $k = 5, 10$. In conclusion, CIP provides very accurate solutions for $k = 2$ but it is slower to solve than MIP(1), while for larger k's MIP(1) dominates in both accuracy and speed.

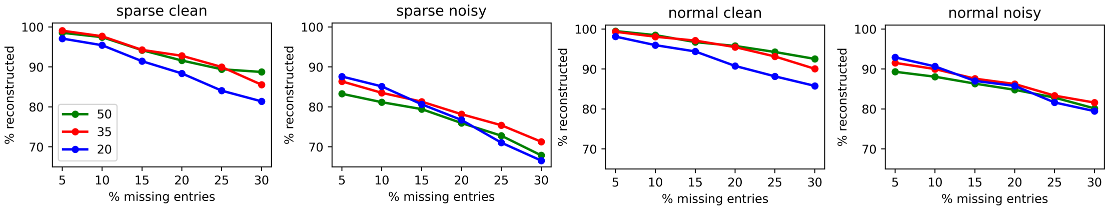
10.2.4 Binary matrix completion

In this section we explore how successful our approach is at recovering missing entries of incomplete binary matrices. We create an incomplete dataset of our synthetic matrices by deleting $5, 10, \dots, 30\%$ of the entries of each matrix. This way, after computing a rank-k factorisation of the incomplete matrix, we can easily compare to the corresponding original matrix to see how many of the entries we have recovered successfully. Since our synthetic matrices are generated to be of Boolean rank at most 10, we cannot expect to recover all the entries by a rank-k completion with $k < 10$ and thus we perform the experiments with $k = 10$.

Figure 10.5 shows the reconstruction percentage against the percentage of missing entries when solving MIP(1) on columns generated by MLP(1) on the incomplete matrices. As previously, the three colours correspond to dimensions of the matrices:

green to 50 × 20, red to 35 × 20 and blue to 20 × 20. We define the percentage of reconstruction as $100 \cdot (1 - \|\mathbf{X} - \mathbf{A} \circ \mathbf{B}\|_F^2 / \|\mathbf{X}\|_F^2)$ where $\mathbf{X}$ is the original complete matrix and $\mathbf{A} \circ \mathbf{B}$ is the rank-k factorisation of the incomplete matrix. As expected the recovery percentage decreases with the percentage of missing entries and clean matrices are better recovered than noisy ones. All in all, we see a very high percentage of the entries can be recovered by MIP(1).

Figure 10.5: Rank-10 binary matrix completion of artificial matrices with $5 - 30\%$ missing entries



10.2.5 Comparing integer programming approaches against heuristics

In this section, we compare our integer programming approaches against the most widely used k-BMF heuristics on real-world datasets. The heuristic algorithms we evaluate include the ASSO algorithm [83, 84], the alternating iterative local search algorithm (ASSO++) of [5] which uses ASSO as a starting point, and the penalty objective formulation (pymf) of [108] via the implementation of [97]. We also compute rank-k NMF, scale rank-1 factors and then binarise them to obtain a k-BMF. The exact details and parameters used in the computations can be found in Appendix B.4.

We solve CIP using CPLEX with a time limit of 20 mins and provide the heuristic solution of k-Greedy as a warm start to it. The column generation approach results are obtained by generating columns for 20 mins using formulation MLP(1) with a warm start of initial rank-1 binary matrices obtained from k-Greedy, then solving MIP(1) over the generated columns with a time limit of 10 mins. Table 10.4 shows the factorisation error in $\| \cdot \|_F^2$ after evaluating the above described methods on all real-world datasets without missing entries for $k = 2, 5, 10$. The best result for each instance is indicated in boldface. We observe that CG provides the strictly smallest error for 8 out of 12 instances.

		MIP(1)	CIP	ASSO++	k-Greedy	pymf	ASSO	NMF
k=2	zoo	272	**271**	276	323	274	367	281
	heart	**1185**	1187	1187	1187	1241	1251	1267
	lymp	1192	**1184**	1202	1201	1225	1352	1272
	apb	**776**	**776**	**776**	**776**	794	778	808
k=5	zoo	**126**	129	133	218	153	354	140
	heart	**737**	738	738	738	813	887	782
	lymp	**982**	1026	1039	1053	1067	1484	1103
	apb	**684**	688	694	688	733	719	721
k=10	zoo	**39**	72	55	175	80	377	51
	heart	425	529	**419**	565	483	694	450
	lymp	**728**	829	812	859	952	1525	821
	apb	**573**	605	591	606	611	661	617

Table 10.4: Comparison of factorisation error in $\|\cdot\|_F^2$ for two IP based methods and five k-BMF heuristics

While integer programming based approaches are able to handle missing entries by simply setting the objective coefficients of the missing entries to 0, the k-BMF heuristics ASSO, ASSO++ and pymf cannot so simply be adjusted. Non-negative matrix factorisation however, has an available implementation that can handle missing entries [69, 70]. Our next experiment compares our integer programming approaches against k-Greedy and NMF on the real datasets that have missing entries. Table 10.5 shows the results with the lowest error results indicated in boldface. For $k = 2$, k-Greedy provides very accurate solutions which MIP(1) and CIP fail to improve on in 3 out of 4 instances. For $k = 5, 10$ however, MIP(1) produces notably lower error factorisations than the other methods.

		MIP(1)	CIP	k-Greedy	NMF
k=2	tumor	**1352**	**1352**	**1352**	1529
	hepatitis	**1264**	1344	1416	1304
	audio	**1419**	**1419**	**1419**	1876
	votes	**1246**	**1246**	**1246**	1268
k=5	tumor	**962**	993	1004	1229
	hepatitis	**1138**	1229	1238	1172
	audio	**1064**	1078	1094	1634
	votes	**779**	853	853	900
k=10	tumor	**514**	632	646	851
	hepatitis	**907**	1048	1056	1013
	audio	**765**	881	881	1580
	votes	**240**	701	706	815

Table 10.5: Comparison of factorisation error in $\|\cdot\|_F^2$ for real-world data with missing entries

Chapter 11

Conclusions

In Part II. of this thesis, we investigated the rank-k binary matrix factorisation problem from an integer programming perspective. We analysed a compact and two exponential size integer programming formulations for the problem and made a comparison on the strength of the formulations' LP-relaxations. We introduced a new objective function, which slightly differs from the traditional squared Frobenius objective in attributing a weight to zero entries of the input matrix that is proportional to the number of times the zero is erroneously covered in a rank-k factorisation. In addition, we discussed a computational approach based on column generation to solve one of the exponential size formulations and reported several computational experiments to demonstrate the applicability of our formulations on real world and artificial datasets.

Our column generation approach is rather computationally challenging and the bottleneck is to compute a tight lower bound on the pricing problem which is needed to determine the master dual bound in Equation (10.1.1). Therefore, it seems that larger datasets are currently out of reach for our methods. If however, one needs an accurate factorisation on moderate size matrices and not a tight optimality gap, our real word data experiments show that our methods provide the lowest error factorisations in most instances with the 600 seconds time limit.

To be able to obtain tighter master dual bounds, future research directions could include developing faster exact algorithms for the pricing problem. In addition, considering semidefinite programming relaxations of the pricing problem to obtain stronger lower bounds could be an interesting avenue to explore. Once, computing good quality lower bounds on the pricing problem is faster, a full branch-and-price implementation would be interesting to explore.

Appendix A

A.1 Minimally non-superfirm interval matrices

We examined several interval matrices with an odd hole in their rectangle cover graph and observed that all of them contained a $\mathbf{D}_4$ submatrix. Therefore, we state the following conjecture.

Conjecture A.1.1. $\mathbf{D}_4$ *is the only minimally non-superfirm interval matrix.*

The below simple result of odd holes in the rectangle cover graph of interval matrices may be useful for the proof of this conjecture. Let us start with an observation.

Observation A.1.2. *Let* $\mathbf{X}$ *be an interval matrix with* $(i_1, j_1), \ldots, (i_4, j_4) \in \operatorname{supp}_1(\mathbf{X})$ *such that the column indices satisfy* $j_1 \leq j_2 \leq j_3 \leq j_4$*. In the rectangle cover graph* $\mathcal{G}(\mathbf{X})$*, if* (i_1, j_1) *is adjacent to* (i_3, j_3) *and* (i_2, j_2) *is adjacent to* (i_4, j_4)*, then* (i_2, j_2) *is adjacent to* (i_3, j_3)*.*

Proof. By the interval property we have

$$\begin{array}{c} \begin{array}{cccc} j_1 & j_2 & j_3 & j_4 \end{array} \\ \begin{bmatrix} x_{i_1,j_1} & \ldots & 1 & \\ & x_{i_2,j_2} & \ldots & 1 \\ 1 & \ldots & x_{i_3,j_3} & \\ & 1 & \ldots & x_{i_4,j_4} \end{bmatrix} \end{array}.$$

If the row or column indices are not distinct then the observation holds in an even simpler way. □

Lemma A.1.3. *Let* C *be an* n*-hole* $n \geq 4$ *in the rectangle cover graph of an interval matrix* $\mathbf{X}'$*. Let* $\mathbf{X}$ *be the submatrix of* $\mathbf{X}'$ *indexed by* $\{i : (i,j) \in C\} \times \{j : (i,j) \in C\}$*.*

(1.) Then $\mathbf{X}$ *has an all* 1*s row.*

(2.) If $\mathbf{X}$ *does not have any repeated rows, then the first and last columns of* $\mathbf{X}$ *both have exactly two* 1*s.*

Proof. (1.) By duplicating rows and columns of $\mathbf{X}$, we may assume that $\mathbf{X}$ is of dimension $n \times n$ and C contains exactly one vertex from each row and column of $\mathbf{X}$. Note that row-column duplication does not alter interval form if the column duplicates are placed directly next to the column they are copied from and it cannot introduce a row of all 1s. Let $v_1, \dots, v_n$ be the vertices of C such that v_j is in column j of $\mathbf{X}$.

First, suppose that v_1 and v_n are not adjacent in $\mathcal{G}(\mathbf{X})$. Let v_ℓ and v_k be the two neighbours of v_1 in C, with $\ell < k < n$. Let T be all the vertices of C which are (1) to the left of v_k and (2) are on a path from v_1 through v_ℓ. Since $k \neq n$, there must exist a vertex $v_t \in T$ which has a neighbour v_p in C with $k < p$. Then we have $v_1, v_t, v_k, v_p \in \operatorname{supp}_1(\mathbf{X})$ with column indices $1 < t < k < p$, hence by Observation A.1.2 v_t is adjacent to v_k in $\mathcal{G}(\mathbf{X})$. However, then $[v_t, v_k]$ is a chord of C which is a contradiction. Therefore, v_1 and v_n must be neighbours in C. Then by the interval form, we have

$$\begin{array}{c} \begin{array}{ccc} 1 & \dots & n \end{array} \\ \begin{bmatrix} v_1 & \dots & 1 \\ 1 & \dots & v_n \end{bmatrix}. \end{array}$$

Therefore, if $v_1 = (i_1, 1)$ and $v_n = (i_n, n)$, rows i_1 and i_n are equal to the all 1s row and the unduplicated form of $\mathbf{X}$ has at least one all 1s row.

(2.) Now let $\mathbf{X}$ not have any row duplicates. Observe that if $\mathbf{X}$ with row duplicates contains a copy of C, then $\mathbf{X}$ without row duplicates also does. By Observation 4.2.1, $\mathbf{X}$ has at least two 1s in each row and column. In addition, by part (1.) of this proof $\mathbf{X}$ has an all 1s row and let this be the first row of $\mathbf{X}$. Let k be the number of columns of $\mathbf{X}$. From part (1.) we know that $(1,1)$ and $(1,k)$ are the leftmost and rightmost vertices of C respectively. Let $v_2, \dots, v_{n-1}$ be the rest of the vertices of $\mathcal{C}$ with v_2 being adjacent to $(1,1)$ and v_{n-1} to $(1,k)$. Then all the rows i that contain vertices $v_3, \dots, v_{n-1}$ must satisfy $x_{i,1} = 0$ as row 1 is an all 1s row and otherwise a chord appears between v_1 and any vertex in row i. Similarly, all the rows i that contain vertices $v_2, \dots, v_{n-2}$ must satisfy $x_{i,1} = 0$. Therefore, column 1 and column k both have exactly two 1s, one in the first row and the second in the row of v_2 and v_{n-1}, respectively. □

Appendix B

B.1 Synthetic data

Table B.1 gives a summary of the parameters used to generate our synthetic dataset. For a synthetic binary matrix **X**, $m \times n$ is the dimension of **X**, κ is the Boolean rank which was used to generate **X**, and $m' \times n'$ is the dimension obtained after removing zero and duplicate row and columns of **X**. Our synthetic data can be downloaded from [59].

(n-sparsity-noise)	$m \times n$	κ	0s%	noise%	#instances	$m' \times n'$
20-sparse-clean	20 × 20	10	75	0	10	14×15
20-sparse-noisy				5		19×19
20-normal-clean			50	0		18×18
20-normal-noisy				5		19×20
35-sparse-clean	35 × 20	10	75	0	10	22×15
35-sparse-noisy				5		31×19
35-normal-clean			50	0		29×18
35-normal-noisy				5		34×20
50-sparse-clean	50 × 20	10	75	0	10	30×15
50-sparse-noisy				5		45×20
50-normal-clean			50	0		40×18
50-normal-noisy				5		48×20

Table B.1: Parameters of the synthetic dataset

B.2 Real world data

Our binarised real world data is available for download at [59]. The following datasets were used in the experiments:

- The Zoo dataset (*zoo*) [35] describes 101 animals with 16 characteristic features. All but one feature is binary. The categorical column which records the number of legs an animal has, is converted into two new binary columns indicating if the number of legs is *less than or equal* or *greater* than four. The size of the resulting fully binary matrix is 101 × 17.

- The Primary Tumor dataset (*tumor*) [58] contains observations on 17 tumour features detected in 339 patients. The features are represented by 13 binary variables and 4 categorical variables with discrete options. The 4 categorical variables are converted into 11 binary variables representing each discrete option. Two missing values in the binary columns are left as missing values. The final dimension of the binary matrix is 339 × 24 with 670 missing values.

- The Hepatitis dataset (*hepat*) [43] consists of 155 samples of medical data of patients with hepatitis. The 19 features of the dataset can be used to predict whether a patient with hepatitis will live or die. 6 of the 19 features take numerical values and are converted into 12 binary features corresponding to options: *less than or equal to the median value*, and *greater than the median value.* The column that stores the sex of patients is converted into two binary columns corresponding to labels man and female. The remaining 12 columns take values *yes* and *no* and are converted into 24 binary columns. The missing values in the raw dataset are left as missing in the binary dataset as well. The final dimension of the binary matrix is 155 × 38 with 334 missing values.

- The SPECT Heart dataset (*heart*) [20] describes cardiac Single Proton Emission Computed Tomography images of 267 patients by 22 binary feature patterns. 25 patients' images contain none of the features and are dropped from the dataset, hence the final dimension of the binary matrix is 242 × 22.

- The Lymphography dataset (*lymp*) [57] contains data about lymphography examination of 148 patients. 8 features take categorical values and are expanded into 33 binary features representing each categorical value. One column is numerical and we convert it into two binary columns corresponding to options: *less than or equal to median value*, and *larger than median value.* The final dimension of the fully binary matrix is 148 × 44.

- The Audiology Standardized dataset (*audio*) [96] contains clinical audiology records on 226 patients. The 69 features include patient-reported symptoms, patient history information, and the results of routine tests which are needed for the evaluation and diagnosis of hearing disorders. 9 features that are categorical valued are binarised into 34 new binary variables indicating if a discrete option is selected. The missing values in the raw dataset are left as missing in the binary dataset as well. The final dimension of the binary matrix is 226×92 with 899 missing values.

- The Amazon Political Books dataset (*books*) [64] contains binary data about 105 US politics books sold by Amazon.com. Columns correspond to books and rows represent frequent co-purchasing of books by the same buyers. The dimension of the binary matrix is 105×105.

- The 1984 United States Congressional Voting Records dataset (*votes*)[98] includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The 16 categorical variables taking values of "voted for", "voted against" or "did not vote", are converted into 16 binary features taking value 1 for "voted for", value 0 for "voted against" and a missing value indicates "did not vote". The final dimension of the binary matrix is 435×16 with 392 missing values.

B.3 Obtaining integer feasible solutions

In this section we give additional numerical results supporting our conclusions drawn in Section 10.2.2.2. Table B.2 shows the factorisation error measured in $\|\cdot\|_F^2$ of integer feasible solutions obtained by solving MIP(1) and MIP$_F$ over columns generated by MLP(1). MIP(1) takes significantly faster to solve than MIP$_F$ but the absolute difference in error between solutions produced by MIP(1) and MIP$_F$ is at most 1, except for the last row in column $k = 5$ where MIP$_F$ runs out of the time budget of 300 seconds and produces higher error solutions than MIP(1).

Table B.3 shows the result of an analogous experiment where the columns used are generated by MLP($\frac{1}{k}$). Since MLP($\frac{1}{k}$) is slower to solve than MLP(1), more columns are generated during CG and the master IPs have a harder task on selecting k columns from a larger set of columns in Table B.3. However, while solving MIP(1) over a larger set of columns adds only a few seconds for most instances, MIP$_F$ runs out of the time budget of 300 secs in about half the cases. This is also demonstrated in the error difference, with solutions by MIP(1) having smaller error than solutions by MIP$_F$ in most cases.

data	k=2		k=5		k=10	
(n-sparsity-noise)	MIP(1)	MIP$_{\mathrm{F}}$	MIP(1)	MIP$_{\mathrm{F}}$	MIP(1)	MIP$_{\mathrm{F}}$
20-sparse-clean	47 (0.0)	47 (0.0)	16 (0.0)	16 (0.0)	0 (0.0)	0 (0.0)
20-sparse-noisy	59 (0.0)	59 (0.0)	30 (0.0)	30 (0.0)	10 (0.0)	10 (0.0)
20-normal-clean	70 (0.0)	**69** (0.3)	27 (0.1)	27 (2.7)	0 (0.0)	0 (0.0)
20-normal-noisy	78 (0.1)	78 (0.9)	40 (0.5)	**39** (76.5)	10 (0.5)	10 (3.4)
35-sparse-clean	84 (0.0)	84 (0.1)	34 (0.0)	34 (0.1)	0 (0.0)	0 (0.0)
35-sparse-noisy	107 (0.0)	107 (0.1)	60 (0.0)	60 (0.6)	23 (0.1)	23 (0.2)
35-normal-clean	125 (0.4)	**124** (2.2)	54 (0.8)	**53** (154.8)	0 (0.0)	0 (0.1)
35-normal-noisy	143 (0.6)	**141** (4.9)	80 (4.1)	80 (245.4)	25 (2.0)	**24** (114.2)
50-sparse-clean	126 (0.0)	126 (0.0)	50 (0.0)	50 (0.1)	0 (0.0)	0 (0.0)
50-sparse-noisy	156 (0.0)	156 (0.1)	89 (0.0)	89 (0.2)	36 (0.0)	36 (0.2)
50-normal-clean	198 (1.4)	**197** (8.2)	91 (30.9)	91 (173.4)	0 (0.1)	0 (0.1)
50-normal-noisy	218 (2.2)	218 (41.4)	**123** (39.7)	126 (271.1)	44 (10.1)	44 (165.8)

Table B.2: Error in $\| \cdot \|_F^2$ (and runtime in seconds) of integer solutions by MIP(1) and MIP$_{\mathrm{F}}$ on columns by MLP(1)

data	k=2		k=5		k=10	
(n-sparsity-noise)	MIP(1)	MIP$_{\mathrm{F}}$	MIP(1)	MIP$_{\mathrm{F}}$	MIP(1)	MIP$_{\mathrm{F}}$
20-sparse-clean	50 (0.0)	50 (0.2)	21 (0.0)	21 (2.6)	0 (0.0)	0 (0.0)
20-sparse-noisy	64 (0.0)	64 (0.6)	**42** (0.1)	43 (219.0)	11 (0.2)	11 (6.3)
20-normal-clean	76 (0.2)	**75** (3.9)	**30** (0.5)	31 (289.6)	0 (0.1)	0 (0.2)
20-normal-noisy	85 (0.3)	85 (6.3)	47 (1.2)	47 (300.4)	11 (0.6)	11 (54.2)
35-sparse-clean	91 (0.0)	91 (1.5)	39 (0.2)	39 (98.9)	0 (0.1)	0 (0.1)
35-sparse-noisy	114 (0.1)	113 (3.1)	**81** (0.5)	84 (300.7)	28 (0.3)	28 (229.9)
35-normal-clean	136 (1.0)	**134** (19.1)	**61** (2.0)	65 (300.8)	0 (0.8)	0 (11.9)
35-normal-noisy	154 (1.6)	154 (58.9)	**93** (6.2)	102 (301.3)	**28** (2.1)	31 (301.0)
50-sparse-clean	137 (0.0)	**136** (0.8)	61 (0.2)	61 (160.0)	0 (0.8)	0 (0.2)
50-sparse-noisy	167 (0.1)	**166** (6.5)	**128** (0.7)	135 (301.5)	**46** (0.6)	50 (301.5)
50-normal-clean	215 (2.2)	215 (131.6)	**100** (34.4)	106 (302.1)	0 (0.8)	0 (153.7)
50-normal-noisy	238 (5.7)	**237** (226.4)	**149** (95.8)	169 (302.9)	**51** (39.4)	62 (302.5)

Table B.3: Error in $\| \cdot \|_F^2$ (and runtime in seconds) of integer solutions by MIP(1) and MIP$_{\mathrm{F}}$ on columns by MLP($\frac{1}{k}$)

B.4 Heuristics for k-BMF

The following methods were evaluated for the comparison in Tables 10.4 and 10.5. Our code is available at [59].

- For the alternating iterative local search algorithm of [5] (ASSO++) we obtained the code from the author's github page, see the reference. The code implements two variants of the algorithm and we report the smaller error solution from two variants of it.

- For the method of [108], we used a python implementation in the package `pymf`, see [97] and we ran it for 10000 iterations.

- We evaluated the heuristic method ASSO [83] which depends on a parameter and we report the best results across nine parameter settings ($\tau \in \{0.1, 0.2, \ldots, 0.9\}$). The code was obtained form the webpage of the author: `https://people.mpi-inf.mpg.de/ pmiettin/src/DBP-progs/`. We observe that ASSO does not return monotone solutions and sometimes we get a higher error solution for a higher value of k.

- In the case of no missing entries in the binary matrix, we used the function `non_negative_factorization` from the `sklearn.decomposition` module in python for the computation of rank-k NMF. We tried all 4 possible initialisation methods: 'nndsvda', 'nndsvd', 'nndsvdar' and 'random'. After obtaining the k-NMF we scale each rank-1 factor to have the same max value on the left and right hand side. Then, we binarise each rank-1 factor with a threshold of $\delta \in \{0.1, 0.2, \ldots, 0.9\}$. In Table 10.4 we report the best result over all these parameter settings.

  As the above python function does not allow missing entries, for incomplete binary matrices we used a Matlab implementation of NMF [70, 69]. Only random initialisation method was available for this implementation and we used 11 different random seeds. Then we performed the same scaling and thresholding as described above and report the best result over all parameter settings in Table 10.5.

- The heuristic k-greedy algorithm was ran with 70 random seeds and the subroutine for BBQP used the greedy and alternating algorithms for BBQP given in Algorithms 3. In addition, the k-greedy algorithm can be run on a preprocessed or original matrix and we tried both ways. For each instance the lowest error factorisation is reported.

References

[1] Christoph Ambhl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM Journal on Computing*, 40:567–596, Jan 2011.

[2] Jerome Amilhastre, M.C. Vilarem, and P. Janssen. Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs. *Discrete Applied Mathematics*, 86(2):125–144, 1998.

[3] Richard P. Anstee and Martin Farber. Characterizations of totally balanced matrices. *Journal of Algorithms*, 5(2):215–230, 1984.

[4] Hans-Jrgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41(2):182–208, 1986.

[5] Francisco Barahona and Joao Goncalves. Local search algorithms for binary matrix factorization, 2019. `https://github.com/IBM/binary-matrix-factorization/blob/master/code`, last accessed on 2020-04-21.

[6] Francisco Barahona and Ali Ridha Mahjoub. On the cut polytope. *Mathematical Programming*, 36(2):157–173, Jun 1986.

[7] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[8] Melanie Beckerleg and Andrew Thompson. A divide-and-conquer algorithm for binary matrix completion. *Linear Algebra and its Applications*, 601:113–133, 2020.

[9] Claude Berge. *Graphs and Hypergraphs.* North Holland, Amsterdam, 1973.

[10] Pierre Bonami, Oktay Günlük, and Jeff Linderoth. Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods. *Mathematical Programming Computation*, 10(3):333–382, Sep 2018.

[11] Andy Boucher. It's hard to color antirectangles. *SIAM Journal on Matrix Analysis and Applications*, 5(2):162–2, Jun 1984.

[12] Andreas Brandstdt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes: A Survey.* Society for Industrial and Applied Mathematics, 1999.

[13] Alberto Caprara and Matteo Fischetti. {0,1/2}-Chvátal-Gomory cuts. *Mathematical Programming*, 74(3):221–235, Sep 1996.

[14] Seth Chaiken, Daniel J. Kleitman, Michael Saks, and James Shearer. Covering regions by rectangles. *SIAM Journal on Algebraic Discrete Methods*, 2(4):394–410, 1981.

[15] Parinya Chalermsook, Sandy Heydrich, Eugenia Holm, and Andreas Karrenbauer. Nearly tight approximability results for minimum biclique cover and partition. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014*, pages 235–246, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[16] Sunil Chandran, Davis Issac, and Andreas Karrenbauer. On the parameterized complexity of biclique cover and partition. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:13, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[17] G. Chaty and Michel Chein. Ordered matching and matchings without alternating cycles in bipartite graphs. *Utilitas Mathematica*, 16:183187, 1979.

[18] Maria Chudnovsky. Berge trigraphs. *Journal of Graph Theory*, 53(1):1–55, 2006.

[19] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *ANNALS OF MATHEMATICS*, 164:51–229, 2006.

[20] Krzysztof J. Cios and Lukasz A. Kurgan. UCI machine learning repository: Spect heart data, 2001. last accessed on 2020-06-11.

[21] Michele Conforti, Gerard Cornuejols, and Giacomo Zambelli. *Integer Programming*. Springer Publishing Company, Incorporated, 2014.

[22] Michele Conforti and M. R. Rao. Structural properties and decomposition of linear balanced matrices. *Mathematical Programming*, 55(1):129–168, 1992.

[23] CPLEX Optimization, Inc., Incline Village, NV. *Using the CPLEX Callable Library, V.12.8*, 2018.

[24] Joseph C. Culberson and Robert A. Reckhow. Covering polygons is hard. *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 601–611, 1988.

[25] William H. Cunningham and Jack Edmonds. A combinatorial decomposition theory. *Canadian Journal of Mathematics*, 32(3):734765, 1980.

[26] Milind Dawande. A notion of cross-perfect bipartite graphs. *Information Processing Letters*, 88(4):143–147, Nov 2003.

[27] Milind Dawande, Pinar Keskinocak, and Sridhar Tayur. On the biclique problem in bipartite graphs. GSIA working paper, Carnegie Mellon University, Pittsburgh, PA 15213, USA, 1996.

[28] Dominique de Caen, David A. Gregory, and Norman J. Pullman. The boolean rank of zero-one matrices. In *Proceedings of the Third Caribbean Conference on Combinatorics and Computing*, pages 169–173, 1981.

[29] Dominique de Caen, David A. Gregory, and Norman J. Pullman. The boolean rank of zero-one matrices II. In *Proceedings of the Vth Caribbean Conference on Combinatorics and Computing*, pages 120–126, 1988.

[30] Caterina De Simone. The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79(1):71 – 75, 1990.

[31] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. last accessed on 2020-06-11.

[32] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 534543, New York, NY, USA, 2002. Association for Computing Machinery.

[33] Samuel Fiorini, Krystal Guo, Marco Macchia, and Matthias Walter. Lower bound computations for the nonnegative rank. In *Proceedings of the 17th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, pages 41–44, Jul 2019.

[34] Herbert Fleischner, Egbert Mujuni, Danil Paulusma, and Stefan Szeider. Covering graphs with few complete bipartite subgraphs. *Theoretical Computer Science*, 410(21):2045 – 2053, 2009.

[35] Richard Forsyth. UCI machine learning repository: Zoo data set, 1990. last accessed on 2020-06-11.

[36] András Frank. Finding minimum generators of path systems. *Journal of Combinatorial Theory, Series B*, 75(2):237244, Mar 1999.

[37] Deborah S. Franzblau and Daniel .J. Kleitman. An algorithm for covering polygons with rectangles. *Information and Control*, 63(3):164–189, 1984.

[38] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[39] Nicolas Gillis and Stephen A. Vavasis. On the complexity of robust PCA and ℓ_1-norm low-rank matrix approximation. *Mathematics of Operations Research*, 43(4):1072–1084, 2018.

[40] Emeric Gioan and Christophe Paul. Split decomposition and graph-labelled trees: Characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Applied Mathematics*, 160(6):708–733, 2012. Fourth Workshop on Graph Classes, Optimization, and Width Parameters Bergen, Norway, October 2009.

[41] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57).* North-Holland Publishing Co., NLD, 2nd edition, 2004.

[42] Martin Charles Golumbic and Clinton F. Goss. Perfect elimination and chordal bipartite graphs. *Journal of Graph Theory*, 2(2):155–163, 1978.

[43] Gail Gong. UCI machine learning repository: Hepatitis data set, 1988. last accessed on 2020-06-11.

[44] David A. Gregory and Norman J. Pullman. Semiring rank: Boolean rank and nonnegative rank factorizations. *Journal of Combinatorics, Information & Systems Sciences*, 8(3):223 – 233, 1983.

[45] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[46] Martin Grötschel, László Lovász, and Alexander Schrijver. Polynomial algorithms for perfect graphs. In C. Berge and V. Chvtal, editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325–356. North-Holland, 1984.

[47] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[48] Hermann Gruber and Markus Holzer. Inapproximability of nondeterministic state and transition complexity assuming P≠NP. In Tero Harju, Juhani Karhumäki, and Arto Lepistö, editors, *Developments in Language Theory*, pages 205–216, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[49] Oktay Günlük, Raphael Andreas Hauser, and Réka Ágnes Kovács. Binary matrix factorization and completion via integer programming. *Mathematics of Operations Research*, 49(2):1278–1302, 2024.

[50] Ervin Győri. A minimax theorem on intervals. *Journal of Combinatorial Theory, Series B*, 37(1):1–9, 1984.

[51] Magns M. Halldrsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45(1):19–23, 1993.

[52] Peter L. Hammer and Frdric Maffray. Completely separable graphs. *Discrete Applied Mathematics*, 27(1):85–99, 1990.

[53] Tao Jiang and B. Ravikumar. Minimal NFA problems are hard. *SIAM Journal on Computing*, 22(6):1117–1141, Dec 1993.

[54] Daniel Karapetyan and Abraham P. Punnen. Heuristic algorithms for the bipartite unconstrained 0-1 quadratic programming problem. arXiv 1210.3684, 2013.

[55] Ki Hang Kim. *Boolean Matrix Theory and Applications.* Monographs and textbooks in pure and applied mathematics. Dekker, 1982.

[56] Ton Kloks and Dieter Kratsch. Computing a perfect edge without vertex elimination ordering of a chordal bipartite graph. *Information Processing Letters*, 55(1):11–16, 1995.

[57] Igor Kononenko and Bojan Cestnik. UCI machine learning repository: Lymphography data set, 1988. last accessed on 2020-06-11.

[58] Igor Kononenko and Bojan Cestnik. UCI machine learning repository.: Primary tumor domain, 1988. last accessed on 2020-06-11.

[59] Réka Á. Kovács. Code for binary matrix factorisation and completion via integer programming, 2021. `https://github.com/kovacsrekaagnes/rank_k_Binary_Matrix_Factorisation`.

[60] Réka Á. Kovács. On minimally non-firm binary matrices. In Ivana Ljubić, Francisco Barahona, Santanu S. Dey, and A. Ridha Mahjoub, editors, *Combinatorial Optimization*, pages 76–88, Cham, 2022. Springer International Publishing.

[61] Réka Á. Kovács, Oktay Günlük, and Raphael A. Hauser. Low-rank boolean matrix approximation by integer programming. In *NeurIPS*, Optimization for Machine Learning Workshop, pages 1–5, Dec 2017. `https://opt-ml.org/papers/OPT2017_paper_34.pdf`.

[62] Réka Á. Kovács, Oktay Günlük, and Raphael A. Hauser. Binary matrix factorisation via column generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):3823–3831, May 2021.

[63] Mehmet Koyutürk and Ananth Grama. PROXIMUS: A framework for analyzing very high dimensional discrete-attributed datasets. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining, KDD '03, pages 147–156, New York, NY, USA, 2003. Association for Computing Machinery.

[64] Valdis Krebs. Amazon political books dataset, 2008. last accessed on 2020-06-11.

[65] Ravi Kumar, Rina Panigrahy, Ali Rahimi, and David Woodruff. Faster algorithms for binary matrix factorization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3551–3559. PMLR, Jun 2019.

[66] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, New York, NY, USA, 1997.

[67] Monique Laurent and Frank Vallentin. Semidefinite optimization, 2016. `https://homepages.cwi.nl/~monique/master_SDP_2016.pdf`, last accessed on 2022-07-9.

[68] Tao Li. A general model for clustering binary data. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 188197, New York, NY, USA, 2005. Association for Computing Machinery.

[69] Yifeng Li and Alioune Ngom. The non-negative matrix factorization toolbox for biological data mining. *Source Code for Biology and Medicine*, 8:10 – 10, 2012.

[70] Yifeng Li and Alioune Ngom. The non-negative matrix factorization toolbox in MATLAB (The NMF MATLAB toolbox), 2013. `https://sites.google.com/site/nmftool/`, last accessed on 2021-07-16.

[71] László Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972.

[72] László Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2(3):253–267, 1972.

[73] László Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979.

[74] Haibing Lu, Jaideep Vaidya, and Vijayalakshmi Atluri. Optimal boolean matrix decomposition: Application to role engineering. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 297–306, Washington, DC, USA, 2008. IEEE Computer Society.

[75] Haibing Lu, Jaideep Vaidya, and Vijayalakshmi Atluri. An optimization framework for role mining. *Journal of Computer Security*, 22(1):1 – 31, 2014.

[76] Anna Lubiw. Doubly lexical orderings of matrices. *SIAM Journal on Computing*, 16(5):854–879, 1987.

[77] Anna Lubiw. The boolean basis problem and how to cover some polygons by rectangles. *SIAM Journal on Discrete Mathematics*, 3(1):98115, Jan 1990.

[78] Anna Lubiw. A weighted min-max relation for intervals. *Journal of Combinatorial Theory, Series B*, 53(2):151–172, 1991.

[79] Christine Lütolf and François Margot. A catalog of minimally nonideal matrices. *Mathematical Methods of Operations Research*, 47(2):221–241, 1998.

[80] Marco E. Lbbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[81] Pasin Manurangsi. Inapproximability of maximum edge biclique, maximum balanced biclique and minimum k-cut from the small set expansion hypothesis. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 79:1–79:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[82] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *Mathematical Programming*, 10(1):147–175, December 1976.

[83] Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases: PKDD 2006*, pages 335–346, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[84] Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, 20(10):1348–1362, Oct 2008.

[85] Pauli Miettinen and Stefan Neumann. Recent developments in boolean matrix factorization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021.

[86] Sylvia D. Monson, Norman J. Pullman, and Rolf Rees. A survey of clique and biclique coverings and factorizations of (0,1)–matrices. *Bulletin – Institute of Combinatorics and its Applications*, 14:17–86, 1995.

[87] Haiko Müller. Alternating cycle-free matchings. *Order*, 7(1):11–21, 1990.

[88] Haiko Mller. On edge perfectness and classes of bipartite graphs. *Discrete Mathematics*, 149(1):159–187, 1996.

[89] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization.* Wiley-Interscience, New York, NY, USA, 1988.

[90] Igor Nor, Danny Hermelin, Sylvain Charlat, Jan Engelstadter, Max Reuter, Olivier Duron, and Marie-France Sagot. Mod/resc parsimony inference: Theory and application. *Information and Computation*, 213:23–32, 2012. Special Issue: Combinatorial Pattern Matching (CPM 2010).

[91] James Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406 – 424, 1977.

[92] Manfred Padberg. The boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical Programming*, 45(1):139–172, August 1989.

[93] Rene Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651 – 654, 2003.

[94] Eric Bruce Phelps. Factor rank of boolean matrices, 1996. PhD Thesis, University of Colorado at Denver.

[95] William R. Pulleyblank. Alternating cycle free matchings. Technical report, CORR 82-18, Department of Combinatorics and Optimization, University of Waterloo, 1982.

[96] Ross Quinlan. UCI machine learning repository: Audiology (standardized) data set, 1992. last accessed on 2020-06-11.

[97] Christopher Schinnerl. Pymf - python matrix factorization module, 2017. `https://github.com/ChrisSchinnerl/pymf3`, last accessed on 2021-03-11.

[98] Jeff Schlimmer. UCI machine learning repository: 1984 US Cong. Voting Records Database, 1987. last accessed on 2020-06-11.

[99] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer Berlin Heidelberg, 2003.

[100] Bao-Hong Shen, Shuiwang Ji, and Jieping Ye. Mining discrete patterns via binary matrix factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 757–766, New York, NY, USA, 2009. Association for Computing Machinery.

[101] Zhongshun Shi, Longfei Wang, and Leyuan Shi. Approximation method to rank-one binary matrix factorization. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 800–805, Aug 2014.

[102] Hans Ulrich Simon. On approximate solutions for combinatorial optimization problems. *SIAM Journal on Discrete Mathematics*, 3:294–310, 1990.

[103] Jinsong Tan. Inapproximability of maximum weighted edge biclique and its applications. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, TAMC'08, page 282293, Berlin, Heidelberg, 2008. Springer-Verlag.

[104] Alan Tucker. A structure theorem for the consecutive 1's property. *Journal of Combinatorial Theory, Series B*, 12(2):153–162, 1972.

[105] Jonathan Wang. A new infinite family of minimally nonideal matrices. *Journal of Combinatorial Theory, Series A*, 118(2):365–372, 2011.

[106] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441 – 466, 1991.

[107] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. Association for Computing Machinery.

[108] Zhongyuan Zhang, Tao Li, Chris Ding, and Xiangsun Zhang. Binary matrix factorization with applications. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM 07, page 391400, USA, 2007. IEEE Computer Society.

[109] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 681690, New York, NY, USA, 2006. Association for Computing Machinery.