Obtaining a Triangular Matrix by Independent Row-Column Permutations

Guillaume Fertin, Irena Rusu, Stéphane Vialette

Obtaining a Triangular Matrix by Independent Row-Column Permutations

Guillaume Fertin[1], Irena Rusu[1], and Stéphane Vialette[2]

[1] LINA UMR CNRS 6241, Université de Nantes, France
{guillaume.fertin,irena.rusu}@univ-nantes.fr
[2] Université Paris-Est, LIGM (UMR 8049), CNRS, UPEM, ESIEE Paris, ENPC, F-77454, Marne-la-Vallée, France
vialette@univ-mlv.fr

Abstract. Given a square $(0, 1)$-matrix A, we consider the problem of deciding whether there exists a permutation of the rows and a permutation of the columns of A such that after carrying out these permutations, the resulting matrix is triangular. The complexity of the problem was posed as an open question by Wilf [7] in 1997. In 1998, DasGupta et al. [3] seemingly answered the question, proving it is **NP**-complete. However, we show here that their result is flawed, which leaves the question still open. Therefore, we give a definite answer to this question by proving that the problem is **NP**-complete. We finally present an exponential-time algorithm for solving the problem.

1 Introduction

In his contribution in the tribute to the late Professor Erdös [7], Wilf posed the following question: *"Let A be a $m \times n$ matrix of 0's and 1's. Consider the computational problem: do there exist permutations P of the rows of A, and Q, of the columns of A such that after carrying out these permutations, A is triangular? The question we ask concerns the complexity of the problem. Is this problem* **NP***-complete? Or, does there exist a polynomial-time algorithm for doing it?"* As noted by Wilf, this problem is strongly related to job scheduling with precedence constraints, a well-known problem in theoretical computer science. The present paper is devoted to giving an answer to this question.

A square matrix is called *lower triangular* if all the entries above the main diagonal are zero. Similarly, a square matrix is called *upper triangular* if all the entries below the main diagonal are zero. A *triangular matrix* is one that is either lower triangular or upper triangular. Because matrix equations with triangular matrices are easier to solve, they are very important in linear algebra and numerical analysis. We refer the reader to [4] for a further discussion.

For an arbitrary square matrix A, it is well-known in linear algebra that there exists an invertible matrix S such that $S^{-1}AS$ is upper triangular. We focus here, however, on permutation matrices. Recall that a permutation matrix is a square matrix obtained from the same size identity matrix by a permutation of rows. A product of permutation matrices is again a permutation matrix and the inverse of a permutation matrix is again a permutation matrix. In fact, for any permutation matrix P, $P^{-1} = P^T$.

This paper is organized as follows. In Section 2, we provide the basic material needed for this paper. Section 3 is devoted to proving hardness of determining whether a square $(0,1)$-matrix is permutation equivalent triangular, i.e. whether it can be transformed into a triangular matrix by independent row and column permutations. In Section 4, we give some properties of permutation equivalent triangular matrices (or pet matrices, for short) and present an exponential-time algorithm to determine whether a matrix is a pet matrix. The paper concludes with suggestions for further research directions.

2 Notations

For any positive integer n, denote $[n] = \{1, 2, \ldots, n\}$. Let $A = [a_{i,j}]$, $1 \leq i \leq m$ and $1 \leq j \leq n$, be a matrix of m rows and n columns. In the case that $m = n$ then the matrix is *square* of *order* n. It is always assumed that the entries of the matrix are elements of some underlying field F. It is convenient to refer to either a row or a column of the matrix as a *line* of the matrix. We use the notation A^T for the *transpose* of matrix A. We always designate a zero matrix by $\mathbf{0}$, a matrix with every entry equal to 1 by J, and the identity matrix of order n by I. In order to emphasize the size of these matrices we sometimes include subscripts. Thus $J_{m,n}$ denotes the all 1's matrix of size m by n, and this is abbreviated to J_n if $m = n$. Notations $\mathbf{0}_{m,n}$, $\mathbf{0}_n$ and I_n are similarly defined. In displaying a matrix we often use $*$ to designate a submatrix of no particular structure. Two matrices A and B are said to be *permutation equivalent*, denoted by $A \sim B$, if there exist permutation matrices P and Q of suitable sizes such that $B = PAQ$.

We will be greatly concerned with matrices whose entries consist exclusively of the integers 0 and 1. Such matrices are referred to as $(0,1)$-*matrices*. For a $(0,1)$-matrix A, we let $\omega(A)$ stand for the number of 1's in A. A square matrix $A = [a_{i,j}]$ of order n is said to be *lower left triangular* if it has only 0's above the main diagonal (*i.e.* $a_{i,j} = 0$ for $1 \leq i < j \leq n$. We write $\triangle_n$ for the lower left triangular $(0,1)$-matrix whose 0's are exclusively above the main diagonal. For two matrices $A = [a_{i,j}]$ and $B = [b_{i,j}]$ of size m by n, we write $A \leq B$ if $a_{i,j} \leq b_{i,j}$ for $1 \leq i \leq m$ and $1 \leq j \leq n$, so that a square matrix A of order n is lower left triangular if $A \leq \triangle_n$. In the context of permutation equivalent matrices, we will sometimes not be interested in any particular orientation of a triangular matrix and forget about any specific orientation such as "*lower left*". Furthermore, for readability, a matrix which is permutation equivalent to a triangular matrix is said to be a *pet matrix*. The *row sum vector* $\mathcal{R}(A) = \begin{bmatrix} r_1 \, r_2 \ldots r_m \end{bmatrix}$ and the *column sum vector* $\mathcal{C}(A) = \begin{bmatrix} c_1 \, c_2 \ldots c_n \end{bmatrix}$ of A are defined by $r_i = \sum_{1 \leq j \leq n} a_{i,j}$ for $1 \leq i \leq m$ and $c_j = \sum_{1 \leq i \leq m} a_{i,j}$ for $1 \leq j \leq n$. The row sum vector $\mathcal{R}(A)$ (resp. column sum vector $\mathcal{C}(A)$) is *stepwise bounded* if $|\{i : r_i \leq k\}| \geq k$ (resp. $|\{j : c_j \leq k\}| \geq k$) for $1 \leq k \leq n$. It is clear that if a $(0,1)$-matrix A is a pet matrix then both $\mathcal{R}(A)$ and $\mathcal{C}(A)$ are stepwise bounded.

The permanent of $A = [a_{i,j}]$ is defined as $\text{per}(A) = \sum_{(j_1,j_2,\ldots,j_n)\in S_n} a_{1,j_1}\, a_{2,j_2}\, \ldots a_{n,j_n}$ where the summation is over all permutations $(j_1, j_2, \ldots, j_n)$ of $[n]$. Observe that, unlike the determinant, we do not put a minus sign in front of some of the terms in the summation. Of particular importance, the permanent does not change if we permute the rows of A and permute the columns of A.

Let $A = [a_{i,j}]$ be an m by n matrix. For convenience, for a set $K \subseteq [m]$ we will write $\overline{K}$ for the set $[m] \setminus K$. Let $K = \{i_1, i_2, \ldots, i_k\}$ be a set of k elements with $K \subseteq [m]$, and let $L = \{j_1, j_2, \ldots, j_l\}$ be a set of l elements with $L \subseteq [n]$. The sets K and L designate a collection of row indices and column indices, respectively, of the matrix A, and the k by l submatrix determined by them is denoted $A[K, L]$.

Let $X = \{x_i : 1 \leq i \leq n\}$ be a non-empty set of n elements, that we call an *n-set*. Let $\mathcal{S} = (S_i : 1 \leq i \leq m)$ be m not necessarily distinct subsets of the n-set X. We refer to this collection of subsets of an n-set as a *configuration of subsets*. We set $a_{i,j} = 1$ if $x_j \in S_i$, and $a_{i,j} = 0$ if $x_i \notin S_i$. The resulting $(0,1)$-matrix $A = [a_{i,j}]$, $1 \leq i \leq m$ and $1 \leq j \leq n$ of size m by n is the *incidence matrix* for the configuration of subsets $\mathcal{S}$ of the n-set X. The 1s in row α_i of A display the elements in the subset S_i, and the 1's in column β_j display the occurrences of x_j among the subsets. Let $\mathcal{S} = (S_i : 1 \leq i \leq n)$ be a configuration of subsets of some ground n-set X. A bijective mapping $\varphi : \mathcal{S} \to [n]$ is said to be a *stepwise bounded labeling* (or sbl for short) of $\mathcal{S}$ if $\left|\bigcup_{\varphi(S_j)\leq i} S_j\right| \leq i$ for $1 \leq i \leq n$.

3 Answering Wilf's question

We prove in this section that, given a square $(0,1)$-matrix A, deciding whether there exists a permutation matrix P and a permutation matrix Q of suitable size such that PAQ is triangular is **NP**-complete.

3.1 Disproving a previous related result

Before giving our proof, it is worth mentioning that the following problem (called LBQIS(n,k) and rephrased to fit the context of this paper) is claimed to be **NP**-complete in [3]: Given a $(0,1)$-matrix of order n and positive integer $k \leq n$, do there exist permutation matrices P and Q such that $PAQ = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$ with $A_{1,2}$ a square lower triangular matrix of size k by k? It is not very difficult to find a polynomial transformation from LBQIS to Wilf's question, which would prove the **NP**-completeness of the latter. Just add $n-k$ empty rows and $n-k$ empty columns to matrix A to obtain a new matrix A'. Now, notice that each submatrix $A_{1,2}$ in a solution for LBQIS may be completed with the $n-k$ empty rows put before row 1 of $A_{1,2}$ and with the $n-k$ empty columns put after column k of $A_{1,2}$ to yield a solution for the instance A' in Wilf's question, and viceversa.

Unfortunately, paper [3] contains a serious flaw in the proof. To fix things, note that in [3] LBQIS is stated in terms of bipartite graphs, for which matrix A is the reduced adjacency matrix. Then, LBQIS(n,k) is proved **NP**-complete by reduction from another problem on bipartite graphs called LBIS(n, k), using the so-called Rearrangement Lemma (Lemma 3.5 in [3]). Two affirmations in the proof of this lemma are contradicted by the following example. Let G be the graph (input for LBIS) with vertices $U = \{i \,|\, 1 \leq i \leq 4\}$ and $V = \{i \,|\, 1 \leq i \leq 4\}$, whose edges are $(1,1), (2,1), (2,2), (3,2), (3,4), (4,3)$ and $(4,4)$. Thus, $n = 4$. Define $k = 1$. Let G' be the input graph for LBQIS built as in [3], and $k' = k^2 + k = 2$. Then the vertex subset $U' \cup V'$ of G', with $U' = \{[2,4],[1,2]\}$ and $V' = \{[1,3],[2,2]\}$ is a solution of LBQIS of size k' for which the assumption on the first line of the Rearrangement Lemma's proof is false. Also, the vertex subset $U' \cup V'$ of G', with $U' = \{[1,1],[2,1]\}$ and $V' = \{[1,2],[1,3]\}$ is a solution of LBQIS of size k' for which the second affirmation in the same lemma ("clearly $q_1 \leq p_1$") is also false.

3.2 Our NP-completeness proof for Wilf's question

We present our results in terms of sbl for configurations of subsets. The rationale for considering sbl for configurations of subsets stems from the following lemma.

Lemma 1. *Let $\mathcal{S} = (S_i : 1 \leq i \leq n)$ be a configuration of subsets of some ground n-set, and let A be the corresponding incidence matrix. There exist permutation matrices P and Q of order n such that $PAQ \leq \triangle_n$ if and only if there exists an sbl of $\mathcal{S}$.*

We need to focus our attention on a special type of sbl. Call a bijective mapping $\varphi : \mathcal{S} \to [n]$ *normalized* if φ maps the identical subsets of elements of $\mathcal{S}$ to a set of consecutive integers. Most of the interest in normalized bijective labelings stems from the following intuitive lemma.

Lemma 2. *Let $\mathcal{S} = (S_i : 1 \leq i \leq n)$ be a configuration of subsets of some ground n-set. If there exists an sbl of $\mathcal{S}$ then there exists a normalized sbl of $\mathcal{S}$.*

We are now ready to prove that deciding whether there exists an sbl of some configuration of subsets is **NP**-complete thereby proving that deciding whether a square $(0,1)$-matrix is a pet matrix is **NP**-complete as well. The proof proceeds by a reduction from the 3SAT problem - which is a known **NP**-complete problem [2]. Let an arbitrary instance of the 3SAT problem be given by a 3CNF formula $\phi = c_1 \vee c_2 \vee \ldots \vee c_m$ over variables $x_1, x_2, \ldots, x_n$. Our construction is divided into two steps: (1) construction of a (polynomial size) ground set $\mathbf{X}$ and (2) construction of a configuration of subsets C of the ground set $\mathbf{X}$. Throughout the proof, parts of the ground set $\mathbf{X}$ are written as capital bold letters $(\mathbf{V}, \mathbf{T}, \mathbf{F}, \ldots)$ and subsets of the configuration are written with capital calligraphic letters $(\mathcal{V}_i, \mathcal{T}_i, \mathcal{F}_i, \ldots)$.

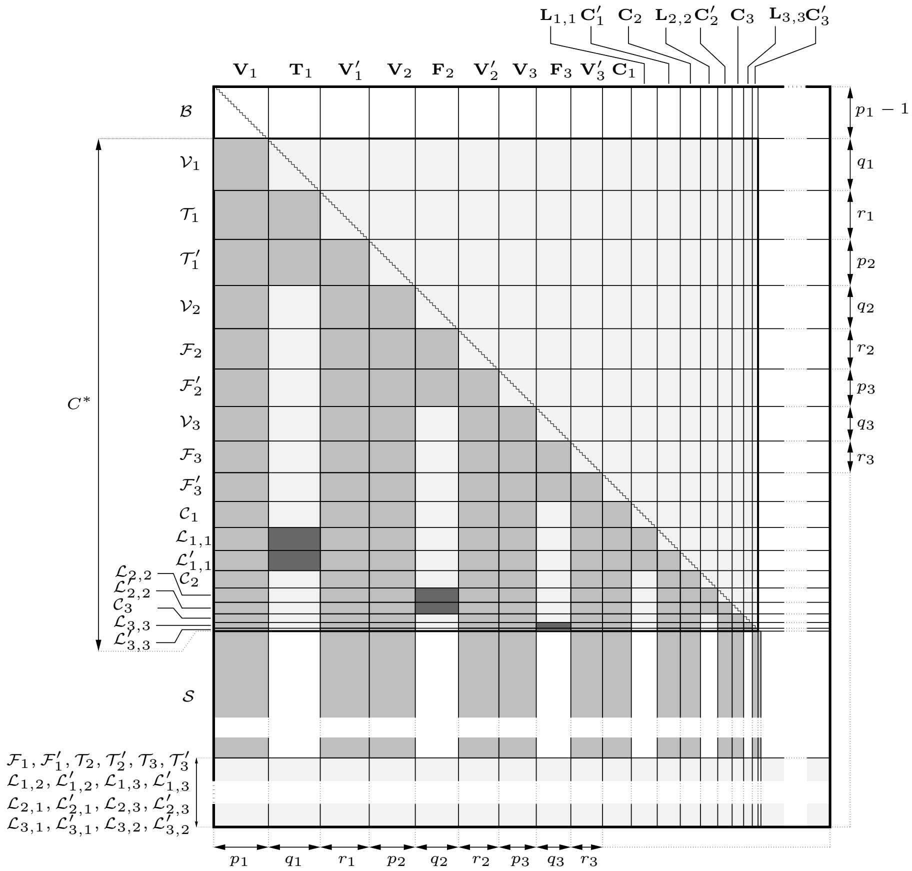

Fig. 1. Illustration of the construction for the 3CNF formula $\phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3)$. Identical subsets are not distinguishable in our representation. A satisfying truth assignment is given by $f(x_1) = \text{TRUE}$, $f(x_2) = \text{FALSE}$ and $f(x_3) = \text{FALSE}$. For sake of clarity, neither the ground set $\mathbf{X}$ nor the collection of subsets C is fully represented.

To begin with, define $p_i = 3(n+m+1-i)+2$, $q_i = 3(n+m+1-i)+1$ and $r_i = 3(n+m+1-i)$ for $1 \leq i \leq n+m$. Furthermore, define $p_{n+m+1} = 1$, $K = \sum_{i=1}^{n} q_i + 2\sum_{i=n+1}^{n+m} q_i$ and $L = \sum_{i=1}^{n+m}(p_{i+1}+r_i)$. Let us now define the ground set $\mathbf{X}$. Consider the pairwise disjoint sets defined as follows: $\mathbf{V}_i = \{v_{i,j} \mid 1 \leq j \leq p_i\}$, $\mathbf{V}'_i = \{v'_{i,j} \mid 1 \leq j \leq r_i\}$, $\mathbf{T}_i = \{t_{i,j} \mid 1 \leq j \leq q_i\}$, $\mathbf{F}_i = \{f_{i,j} \mid 1 \leq j \leq q_i\}$ for $1 \leq i \leq n$. Furthermore, define $\mathbf{C}_i = \{c_{i,j} \mid 1 \leq j \leq p_{n+i}\}$ $\mathbf{C}'_i = \{c'_{i,j} \mid 1 \leq j \leq r_{n+i}\}$ for $1 \leq i \leq m$, and $\mathbf{L}_{i,k} = \{\ell_{i,k,j} \mid 1 \leq j \leq q_{n+i}\}$ for $1 \leq i \leq m$ and $1 \leq k \leq 3$. Finally, define $\mathbf{S} = \{s\}$. For simplicity of notation, write $\mathbf{V} = \bigcup_{1 \leq i \leq n} \mathbf{V}_i$, $\mathbf{V}' = \bigcup_{1 \leq i \leq n} \mathbf{V}'_i$, $\mathbf{T} = \bigcup_{1 \leq i \leq n} \mathbf{T}_i$, $\mathbf{F} = \bigcup_{1 \leq i \leq n} \mathbf{F}_i$, $\mathbf{C} = \bigcup_{1 \leq i \leq m} \mathbf{C}_i$, $\mathbf{C}' = \bigcup_{1 \leq i \leq m} \mathbf{C}'_i$, and $\mathbf{L}_i = \bigcup_{1 \leq k \leq 3} \mathbf{L}_{i,k}$ for $1 \leq i \leq m$ and $\mathbf{L} = \bigcup_{1 \leq i \leq m} \mathbf{L}_i$. Informally, elements of $\mathbf{V} \cup \mathbf{V}'$ are associated to variables, elements of $\mathbf{T} \cup \mathbf{F}$ are associated to literals, elements of $\mathbf{C} \cup \mathbf{C}'$ are associated to clauses, elements of $\mathbf{L}$ are associated to literals in clauses and $\mathbf{S}$ is a *separator set*. The ground set $\mathbf{X}$ of our construction is defined to be $\mathbf{X} = \mathbf{V} \cup \mathbf{V}' \cup \mathbf{T} \cup \mathbf{F} \cup \mathbf{C} \cup \mathbf{C}' \cup \mathbf{L} \cup \mathbf{S}$.

Having defined the ground set $\mathbf{X}$, we now turn to the detailed construction of a configuration of subsets C of $\mathbf{X}$. For sake of clarity, this will be divided into several steps. First, each variable x_i, $1 \leq i \leq n$, is associated to identical subsets $\mathcal{V}_{i,j}$, $1 \leq j \leq q_i$, in C. These subsets are defined as follows: $\mathcal{V}_{i,j} = \left(\bigcup_{1 \leq k \leq i} \mathbf{V}_k\right) \cup \left(\bigcup_{1 \leq k \leq i-1} \mathbf{V}'_k\right)$ for $1 \leq i \leq n$ and $1 \leq j \leq q_i$. Let us denote by $\mathcal{V}_i$, $1 \leq i \leq n$, the collection $(\mathcal{V}_{i,j} \mid 1 \leq j \leq q_i)$. Next, each (positive) literal x_i, $1 \leq i \leq n$, is associated to identical subsets $\mathcal{T}_{i,j}$, $1 \leq j \leq r_i$, and to identical subset $\mathcal{T}'_{i,j}$, $1 \leq j \leq p_{i+1}$. These subsets are defined as follows: $\mathcal{T}_{i,j} = \mathbf{T}_i \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{V}_k\right) \cup \left(\bigcup_{1 \leq k \leq i-1} \mathbf{V}'_k\right)$ for $1 \leq i \leq n$ and $1 \leq j \leq r_i$, and $\mathcal{T}'_{i,j} = \mathbf{T}_i \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{V}_k\right) \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{V}'_k\right)$ for $1 \leq i \leq n$ and $1 \leq j \leq p_{i+1}$. Of course, a similar construction of subsets applies for the negation $\overline{x}_i$ of each variable x_i, *i.e.*, $\mathcal{F}_{i,j} = \mathbf{F}_i \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{V}_k\right) \cup \left(\bigcup_{1 \leq k \leq i-1} \mathbf{V}'_k\right)$ for $1 \leq i \leq n$ and $1 \leq j \leq r_i$, and $\mathcal{F}'_{i,j} = \mathbf{F}_i \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{V}_k\right) \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{V}'_k\right)$ for $1 \leq i \leq n$ and $1 \leq j \leq p_{i+1}$. For readability, write $\mathcal{T}_i = (\mathcal{T}_{i,j} \mid 1 \leq j \leq r_i)$, $\mathcal{T}'_i = (\mathcal{T}'_{i,j} \mid 1 \leq j \leq p_{i+1})$, $\mathcal{F}_i = (\mathcal{F}_{i,j} \mid 1 \leq j \leq r_i)$ and $\mathcal{F}'_i = (\mathcal{F}'_{i,j} \mid 1 \leq j \leq p_{i+1})$ for $1 \leq i \leq n$. Note that the following (strict) inclusions hold for all $1 \leq i \leq n$, $1 \leq j_1 \leq q_i$, $1 \leq j_2 \leq r_i$ and $1 \leq j_3 \leq p_{i+1}$: (i) $\mathcal{V}_{i,j_1} \subset \mathcal{T}_{i,j_2} \subset \mathcal{T}'_{i,j_3}$ and (ii) $\mathcal{V}_{i,j_1} \subset \mathcal{F}_{i,j_2} \subset \mathcal{F}'_{i,j_3}$. We now turn to the m clauses of the 3CNF formula. Each clause c_i, $1 \leq i \leq m$, is associated to identical subsets $\mathcal{C}_{i,j}$, $1 \leq j \leq q_{n+i}$. These subsets are defined as follows: $\mathcal{C}_{i,j} = \mathbf{V} \cup \mathbf{V}' \cup \left(\bigcup_{1 \leq k \leq i} \mathbf{C}_k\right) \cup \left(\bigcup_{1 \leq k \leq i-1} \mathbf{C}'_k\right)$ for $1 \leq i \leq m$ and $1 \leq j \leq q_{n+i}$. Let us denote by $\mathcal{C}_i$, $1 \leq i \leq m$, the collection $(\mathcal{C}_{i,j} \mid 1 \leq j \leq q_{n+i})$. It is easily seen that $\mathcal{V}_{i,j_1} \subset \mathcal{C}_{k,j_2}$ for all $1 \leq i \leq n$, $1 \leq j_1 \leq q_i$, $1 \leq k \leq m$ and $1 \leq j_2 \leq q_{n+k}$.

Now, we consider the only part of the construction that depends on which literal occurs in which clauses. Denote by $\lambda_{i,k}$ the k-th literal of clause c_i, that is write $c_i = \lambda_{i,1} \vee \lambda_{i,2} \vee \lambda_{i,3}$ for $1 \leq i \leq m$, where each $\lambda_{i,k}$ is a variable or its negation. The k-th literal, $1 \leq k \leq 3$, of each clause c_i, $1 \leq i \leq m$, is associated to identical subsets $\mathcal{L}_{i,k,j}$, $1 \leq j \leq r_{n+i}$, and to identical subsets $\mathcal{L}'_{i,k,j}$, $1 \leq j \leq p_{n+i+1}$. These subsets are defined as follows: $\mathcal{L}_{i,k,j} = \mathbf{V} \cup \mathbf{V}' \cup \mathbf{A}_k \cup \mathbf{L}_{i,k} \cup \left(\bigcup_{1 \leq \ell \leq i} \mathbf{C}_\ell\right) \cup \left(\bigcup_{1 \leq \ell \leq i-1} \mathbf{C}'_\ell\right)$ for $1 \leq i \leq m$, $1 \leq j \leq r_{n+i}$ and $1 \leq k \leq 3$ and $\mathcal{L}'_{i,k,j} = \mathbf{V} \cup \mathbf{V}' \cup \mathbf{A}_k \cup \mathbf{L}_{i,k} \cup \left(\bigcup_{1 \leq \ell \leq i} \mathbf{C}_\ell\right) \cup \left(\bigcup_{1 \leq \ell \leq i} \mathbf{C}'_\ell\right)$ for $1 \leq i \leq m$, $1 \leq j \leq p_{n+i+1}$ and $1 \leq k \leq 3$, where $\mathbf{A}_k = \mathbf{T}_\ell$ if $\lambda_{i,k} = x_\ell$ and $\mathbf{A}_k = \mathbf{F}_\ell$ if $\lambda_{i,k} = \overline{x}_\ell$. For the sake of clarity, write $\mathcal{L}_{i,k} = (\mathcal{L}_{i,k,j} \mid 1 \leq j \leq r_{n+i})$ and $\mathcal{L}'_{i,k} = (\mathcal{L}'_{i,k,j} \mid 1 \leq j \leq p_{n+i+1})$ for $1 \leq i \leq m$ and $1 \leq k \leq 3$. Again, observe that $\mathcal{C}_{i,j_1} \subset \mathcal{L}_{i,k,j_2} \subset \mathcal{L}_{i,k,j_2}$ for all $1 \leq i \leq m$, $1 \leq j_1 \leq q_{n+i}$, $1 \leq j_2 \leq r_{n+i}$, $1 \leq j_3 \leq p_{n+i+1}$ and $1 \leq k \leq 3$.

Our construction ends with $p_1 + K - 1$ *utility subsets*. These subsets will be partitioned into two separate classes according to their indented function:

bootstrap subsets and *separator subsets*. First, C contains identical bootstrap subsets $\mathcal{B}_i$, $1 \leq i \leq p_1 - 1$, defined as follows: $\mathcal{B}_i = \emptyset$ for $1 \leq i \leq p_1 - 1$. The idea is to force any sbl to map the $p_1 - 1$ empty sets of $\mathcal{B}$ to the first $p_1 - 1 = 3(n + m) + 1$ integers. Indeed, it is easily seen that all the above defined subsets of the configuration of subsets C but those of $\mathcal{B}$ contain at least p_1 elements and hence cannot be mapped to an integer $i \leq p_1 - 1$ in any sbl of C. Second, C contains identical separator subsets $\mathcal{S}_i$, $1 \leq i \leq K$, defined by: $\mathcal{S}_i = \mathbf{V} \cup \mathbf{V}' \cup \mathbf{C} \cup \mathbf{C}' \cup \mathbf{S}$ for $1 \leq i \leq K$. The rationale of these subsets is that we need a separator between subsets in C corresponding to a satisfying truth assignment f for the 3CNF formula ϕ and garbage subsets of C, that is subsets not involved in the satisfying truth assignment f. For simplicity, let us denote by $\mathcal{B}$ the collection $(\mathcal{B}_i \mid 1 \leq i \leq p_1 - 1)$ and by $\mathcal{S}$ the collection $(\mathcal{S}_i \mid 1 \leq i \leq K)$. Clearly our construction can be carried on in polynomial time: indeed, we have $|\mathbf{X}| = O(m^2 + n^2)$ and $|C| = O(m^2 + n^2)$.

Lemma 3. *There exists a satisfying truth assignment f for ϕ if and only if there exists an sbl of the configuration of subsets C of the ground set $\mathbf{X}$.*

The key elements of the proof are as follows. First, it is crucial to focus on solutions that map identical subsets of elements of $\mathcal{S}$ to a set of consecutive elements (see Lemma 2). Second, the general shape of the solution is largely guided by the construction. Indeed, the empty subsets have to be placed first, followed by subsets corresponding to literals (either the positive or the negative literal of each variable has been chosen) and next by subsets corresponding to clauses (one satisfying literal of each clause is chose). Finally the separator subsets have to be placed, with the result that (thanks to the large polynomial number of such subsets) the remaining subsets can be placed in any order without violating the sought sbl property. The reader is invited to consider Figure 1 for a schematic illustration of the reduction. We now briefly discuss, in an informal way, the two key arguments that are used in the proof. First, the whole procedure is, to some extent, similar to the accounting method used in amortized complexity analysis. Indeed, one might view the operation of placing a set (one after the other) as the process of charging some customer, the cost being the number of new elements that are introduced. With this metaphor in mind, notice that we do not charge when a subset does not introduce any new element, so that the leftover amount can be stored as "credit". When we place a new subset that does introduce some new elements, we can use the "credit" stored to pay for the cost of the operation. Second, when a subset uses the "credit" stored to pay the cost of introducing new elements, the following invariants can be shown to hold true: (i) it uses all the available credit and (ii) it does not allow to accumulate (it should be now clear that consecutive identical subsets do allow for accumulating credit) as much credit as it has consumed, thereby proving that subsets introduce less and less new elements as we progress adding subsets one after the other.

The main result of this paper can now be stated.

Theorem 1. *Let A be a $(0,1)$-matrix. Deciding whether A is a pet matrix is* **NP**-*complete.*

4 Exponential-time algorithm

We present here an exponential-time algorithm for deciding whether a given a $(0,1)$-matrix A of order n is a pet matrix. We start by presenting some basic properties of square $(0,1)$-matrices that can be transformed into some triangular matrix by row and column independent permutations to help solving involved algorithmic issues. We of course focus of polynomial-time checkable properties.

We first focus on the permanent of a square $(0,1)$-matrix. A well-known result (see e.g. [1]) states that for a $(0,1)$-matrix A of order n, one has $\mathrm{per}(A) = 1$ if and only if the lines of A may be permuted to yield a triangular matrix with 1's in the n main diagonal positions and 0's above the mail diagonal. This theorem amounts to saying that $\mathrm{per}(A) = 1$ if and only if there exist permutation matrices P and Q such that $I \leq PAQ \leq$ ◺. As shown in the following lemma, $\mathrm{per}(A) = 1$ is certainly a threshold value in our context.

Lemma 4. *Let A be $(0,1)$-matrix. If A is a pet matrix then* $\mathrm{per}(A) \leq 1$.

Notice that deciding $\mathrm{per}(A) \leq 1$ for $(0,1)$-matrices of order n reduces to computing at most $n+1$ perfect matchings in bipartite graphs [1], and hence the above test is $O(n^3\sqrt{n})$ time as the Hopcroft–Karp algorithm for computing a maximum matching in a bipartite graph $B = (V,E)$ runs in $O(|E|\sqrt{|V|})$ [6].

Next, it is a simple matter to check that if a $(0,1)$-matrix A of order n is a pet matrix, then it contains at most $\frac{1}{2}n(n+1)$ 1's (*i.e.*, $\omega(A) \leq \frac{1}{2}n(n+1)$). The following lemma gives a lower bound.

Lemma 5. *Let A be $(0,1)$-matrix of order n, $n \geq 2$. If A contains at most $n+1$ 1's, then A is a pet matrix.*

Notice that, albeit not very impressive, Lemma 5 is tight as the square matrix $\begin{bmatrix} I_{n-2} & \mathbf{0}_{n-2,2} \\ \mathbf{0}_{2,n-2} & J_2 \end{bmatrix}$ of order n has $n-2+4 = n+2$ 1's and is not a pet matrix.

Finally, the following trivial lemma gives another condition that helps improving the running time of the algorithm in practice.

Lemma 6. *Let A be $(0,1)$-matrix of order n and D the directed graph associated to A (*i.e.*, the adjacency matrix of D is A). If the digraph D is acyclic (regardless self-loops), then A is a pet matrix.*

We now turn to presenting the exponential-time algorithm. The simplest exhaustive algorithm considers every possible pairs of permutation matrices (P,Q) yielding a $O((n!)^2 \cdot \mathrm{poly}(n))$ time algorithm. However, according to Lemma 1, it is enough to consider every permutation matrix P of order n and check whether the first i, $1 \leq i \leq n$, rows of PA have 1's in at most i columns. This observation yields a $O(n! \cdot \mathrm{poly}(n))$ time algorithm. We propose here another exhaustive algorithm that improves on the $O(n! \cdot \mathrm{poly}(n))$ time algorithm. The basic idea is to recursively split into smaller submatrices, instead of enumerating all permutations. For a $(0,1)$-matrix A of order n, we consider every possible set R of $\lceil n/2 \rceil$ rows of A and every possible set of $\lceil n/2 \rceil$ columns C of A, and check

$$PAQ = \begin{bmatrix} A_1 & \mathbf{0} \\ * & A_2 \end{bmatrix}$$
(a) Even

$$PAQ = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} \\ * & 1 & \mathbf{0} \\ * & * & A_2 \end{bmatrix}$$
(b) Odd and one 1

$$PAQ = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} \\ * & 0 & \mathbf{0} \\ * & * & A_2 \end{bmatrix}$$
(c) Odd and zero 1

Fig. 2.

whether these lines induce a zero matrix (or a matrix with at most one 1 in case the matrix has odd order; details follow).

If n is even, we let P and Q be two permutation matrices that put the rows in R at the first $\lceil n/2 \rceil$ positions and the columns in C at the last $\lceil n/2 \rceil$ positions. The key element for the improvement is that no specific order is required for the rows in R nor for the columns in C. The algorithm rejects the matrix A for the subsets R and C if $\omega(A[R,C]) > 1$, otherwise we can write PAQ as in Figure 2(a), where A_1 and A_2 are matrices of order $\lceil n/2 \rceil = n/2$, and we proceed to recursively check that both A_1 and A_2 are pet matrices.

The case n is odd is a little bit more complicated. First, the algorithm rejects matrix A for the subsets R and C if $\omega(A[R,C]) > 1$. Otherwise, we need to consider two (possibly positive) cases: (i) $\omega(A[R,C]) = 1$ or (ii) $\omega(A[R,C]) = 0$. If $\omega(A[R,C]) = 1$, we let P and Q be two permutation matrices that put the rows in R at the first $\lceil n/2 \rceil$ positions and the columns of C at the last $\lceil n/2 \rceil$ positions (no specific order for the rows in R nor for the columns in C, except that the 1 of A is at row index $\lceil n/2 \rceil$ and at column index $\lceil n/2 \rceil$ in PAQ). We can write PAQ as in Figure 2(b), where A_1 and A_2 are matrices of order $\lceil n/2 \rceil - 1 = n/2$, and we proceed to recursively check that both A_1 and A_2 are pet matrices. Finally, if $\omega(A[R,C]) = 0$, for every row index $i \in R$ and every column index $j \in C$, we let P and Q be two permutation matrices that put the rows in R at the first $\lceil n/2 \rceil$ positions and the columns of C at the last $\lceil n/2 \rceil$ positions (no specific order for the rows in R nor for the columns in C except that row i in A is at row index $\lceil n/2 \rceil$ and column index j is at column index $\lceil n/2 \rceil$ in PAQ). We can write PAQ as in Figure 2(c), where A_1 and A_2 are matrices of order $\lceil n/2 \rceil - 1 = n/2$, and we proceed to recursively check that both A_1 and A_2 are pet matrices.

A detailed description is given in Algorithms 1, 2 and 3. We now turn to evaluating the time complexity of this algorithm and we write $T(n)$ for the time complexity of calling $\mathsf{permTriangular}(A)$ for some $(0,1)$-matrix A or order n.

$$T(n) \leq \begin{cases} \left(\lceil n/2 \rceil\right)^2 \binom{n}{\lceil n/2 \rceil}^2 \left(2T\left(\lfloor n/2 \rfloor\right) + 1\right) + O(n^3\sqrt{n}) & \text{if } n \text{ is odd} \\ 2\left(\lceil n/2 \rceil\right)^2 \binom{n}{\lceil n/2 \rceil}^2 T\left(\lfloor n/2 \rfloor\right) + O(n^3\sqrt{n}) & \text{if } n \text{ is odd} \end{cases}$$

with $T(1) = O(1)$. The $O(n^3\sqrt{n})$ term is the time complexity for lines 2 and 3 in Algorithm 1. We also observe that the worst case occurs when $n = 2^m - 1$ as $\lceil n/2 \rceil, \lceil n/4 \rceil, \ldots$ are odd integers. Looking for an asymptotic solution of the worst case, we thus write the following simplified recurrence: $T(2^m) =$

Algorithm 1: Recognizing pet matrices.

Algorithm: permTriangular

Data: A square matrix $A = [a_{i,j}]$ of order n
Result: true if A is a pet matrix, false otherwise

if ($\omega(A) \leq n+1$) *or* ($\mathrm{per}(A) = 1$) *or (A is stepwise bounded) or (*$D(A)$ *is acyclic)* **then return** true

if ($\omega(A) > \frac{n(n+1)}{2}$) *or* ($\mathrm{per}(A) > 1$) *or (*$\mathcal{R}(A)$ *or* $\mathcal{C}(A)$ *is not stepwise bounded)* **then return** false

for *every subset* $R \subset [n]$ *of size* $\lceil \frac{n}{2} \rceil$ *and every subset* $C \subset [n]$ *of size* $\lceil \frac{n}{2} \rceil$ **do**
  if n *is even* **then**
    return permTriangularEven(A, R, C)
  else
    return permTriangularOdd(A, R, C)

return false

Algorithm 2: Subprocedure for recognizing pet matrices of even order.

Algorithm: permTriangularEven

Data: A square matrix $A = [a_{i,j}]$ of even order n, and non-empty subsets $R \subset [n]$ and $C \subset [n]$, both of size $\frac{n}{2}$
Result: true if A is a pet matrix with $A[R, C]$ as the upper right submatrix, false otherwise

if $\omega(A[R, C]) > 0$ **then return** false

Let $A_{\mathrm{ul}} = A[R, \overline{C}]$ and $A_{\mathrm{lr}} = A[\overline{R}, C]$

return permTriangular(A_{ul}) && permTriangular(A_{lr})

$2^{2m-2} \binom{2^m}{2^{m-1}}^2 \left(2T(2^{m-1}) + 1\right) + 2^{7m/6}$, with $T(1) = 1$. Now, write $\alpha(2^m) = 2^{2m-2} \binom{2^m}{2^{m-1}}^2$. Clearly, $\alpha(2^m) \geq 2^{7m/6}$, and hence we focus for now on on the recurrence $T(2^m) = 2\,\alpha(2^m)\,\left(T(2^{m-1}) + 1\right)$. A convenient non-recursive form of $T(2^m)$ is given in the following lemma.

Lemma 7. $T(2^m) = \left(2^m \prod_{i=1}^{m} \alpha(2^i)\right) + \left(\sum_{i=1}^{m} 2^{m-i+1} \prod_{j=i}^{m} \alpha(2^j)\right)$.

We now need the following lemma, in order to give an asymptotic solution for $T(n)$ in Proposition 1.

Lemma 8. $\sum_{i=1}^{m} 2^{m-i} \prod_{j=i}^{m} \alpha(2^j) = O\left(m\, 2^{2^{m+2}+m+1}\right)$.

Proposition 1. *Algorithm* permTriangular *runs in* $O\left(n\, 2^{4n}\, \pi^{-\log(n)}\right)$ *time.*

Proof. We have already observed that the worst case occurs for $n = 2^m - 1$. According to Lemma 8, we have $T(2^m) = O\left(2^{2^{m+2}+m-3}\, \pi^{-m}\right)$ and hence $T(n) = O\left(2^{2^{\log(n)+2}+\log(n)-3}\, \pi^{-\log(n)}\right) = O\left(n\, 2^{4n}\, \pi^{-\log(n)}\right)$. □

Algorithm 3: Subprocedure for recognizing pet matrices of odd order.

Algorithm: permTriangularOdd

Data: A square matrix $A = [a_{i,j}]$ of odd order n, and non-empty subsets $R \subset [n]$ and $C \subset [n]$, both of size $\lceil \frac{n}{2} \rceil$

Result: true if A is a pet matrix with $A[R, C]$ as the upper right submatrix

if $\omega(A[R, C]) > 1$ **then return** false
if $\omega(A) = 0$ **then**
  for *every* $i \in R$ *and every* $j \in C$ **do**
    Let $A_{\mathrm{ul}} = A[R \setminus \{i\}, \overline{C}]$ and $A_{\mathrm{lr}} = A[\overline{R}, C \setminus \{j\}]$
    if permTriangular(A_{ul}) && permTriangular(A_{lr}) **then return** true
  return false
else
  Let i and j be the row and column indices of the unique 1 in $A[R, C]$
  Let $A_{\mathrm{ul}} = A[R \setminus \{i\}, \overline{C}]$ and $A_{\mathrm{lr}} = A[\overline{R}, C \setminus \{j\}]$
  return permTriangular(A_{ul})) && permTriangular(A_{lr})

5 Conclusion

We suggest for further research directions regarding the hardness of recognizing pet $(0, 1)$-matrices. (i) Suppose a $(0, 1)$-matrix A of order n has $n + k$ 1's with $n + 2 \leq n + k \leq \frac{1}{2}n(n + 1)$. Can one decide in $f(k)\, n^{O(1)}$ time whether A is a pet matrix, where f is an arbitrary function depending only on k? (ii) What is the average running time of Algorithm permTriangular for pet matrices? (iii) A graph labeling strongly related to symmetric pet $(0, 1)$-matrices can be defined as follows: Given a graph $G = (V, E)$ or order n, decide whether there exists a bijective mapping $f : V \to [n]$ such that $f(u) + f(v) > n$ for every edge $\{u, v\} \in E$ (*i.e.*, $PAP^T \leq \triangle_n$). Investigating the relationships between the two combinatorial problems is expected to yield fruitful results.

References

1. R.A. Brualdi and H.J. Ryser. *Combinatorial Matrix Theory.* Cambridge University Press, New York, 1991.
2. S.A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd Annual ACM Symposium on Theory of Comput.*, pages 151–158, New York, 1971. ACM.
3. B. DasGupta, T. Jiang, S. Kannan, M. Li, and E. Sweedyk. On the complexity and approximation of syntenic distance. *Discrete Applied Mathematics*, 88(1-3):59–82, 1998.
4. G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore and London, 3rd edition, 1996.
5. R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics.* Addison-Wesley, second edition, 1994.
6. J.E. Hopcroft and R.M. Karp. An $O(n^{2.5})$ algorithm for matching in bipartite graphs. *SIAM Journal on Computing*, 4:225–231, 1975.

7. H.S. Wilf. On crossing numbers, and some unsolved problems. In B. Bollobás and A. Thomason, editors, *Combinatorics, Geometry and Probability: A Tribute to Paul Erdös*, pages 557–562. Cambridge University Press, 1997.

Appendix (Reviewers' version only)

Proof (of Lemma 1). The forward direction if obvious. For the reverse direction, let $\mathcal{S} = (S_i : 1 \leq i \leq n)$ be a configuration of subsets of some ground set of cardinality n and $\varphi : \mathcal{S} \to [n]$ be a bijective mapping such that $\left|\bigcup_{S \in \mathcal{S}:\varphi(S) \leq i} S\right| \leq i$ for all $1 \leq i \leq n$. Let $A = [a_{i,j}]$ be the incidence matrix of $\mathcal{S}$. Now let P be the permutation matrix of order n that permutes the rows of A so that row i of PA correspond to subset $\varphi^{-1}(i)$ for all $1 \leq i \leq n$. We denote this row-permuted matrix by $A' = [a'_{i,j}]$. Define the function $\tau : [n] \to [n+1]$ that indicates for each column index j the minimum row index i so that $a'_{i,j} = 1$, and we adhere to the convention that $\tau(j) = n+1$ if row j does not contain a 1. Define a permutation matrix Q or order n that permutes the columns of A' by ascending τ values, breaking ties arbitrarily. We designate this row-permuted matrix by $A'' = [a''_{i,j}]$. We claim that $A'' = PAQ \leq \mathbin{\triangle}_n$. Indeed, suppose, aiming at a contradiction, that $A'' \not\leq \mathbin{\triangle}_n$. Let $i_{\min}$ be the smallest row index such that $a''_{i_{\min},j_{\min}} = 1$ for some $j_{\min} > i_{\min}$. Then it follows that $\tau(j) \leq \tau(j_{\min})$ for $j < j_{\min}$ and hence $\left|\bigcup_{S \in \mathcal{S}:\varphi(S) \leq i_{\min}} S\right| > i_{\min}$. This is the desired contradiction, and hence $A'' = PAQ \leq \mathbin{\triangle}_n$. □

Proof (of Lemma 2). The proof is by contradiction. Denote by $\Phi(\mathcal{S})$ the set of all sbl s of the configuration of subsets $\mathcal{S}$. We claim that there exists a normalized sbl $\varphi \in \Phi(\mathcal{S})$ which maps the identical subsets of $\mathcal{S}$ to sets of consecutive integers. For each $\varphi \in \Phi(\mathcal{S})$, define

$$\mathcal{M}(\varphi) = \{S_j \mid \exists S_i, S_k \text{ s.t. } \varphi(S_i) < \varphi(S_k) < \varphi(S_j) \text{ and } S_i = S_j \neq S_k\}$$

Then, there exists a mapping $\varphi^* \in \Phi(\mathcal{S})$ such that $|\mathcal{M}(\varphi^*)| \leq |\mathcal{M}(\varphi')|$ for all $\varphi' \in \Phi(\mathcal{S})$. We show that $|\mathcal{M}(\varphi^*)| = 0$, and hence that φ^* is our desired mapping. Suppose, for the sake of contradiction, that $|\mathcal{M}(\varphi^*)| > 0$. Let $S_j \in \mathcal{M}(\varphi^*)$ be such that $\varphi^*(S_j) \leq \varphi^*(S)$ for all $S \in \mathcal{M}(\varphi^*)$. Then, there exists two subsets $S_i, S_k \in \mathcal{S}$ such that $S_i = S_j \neq S_k$ and $\varphi^*(S_i) + 1 = \varphi^*(S_k) < \varphi^*(S_j)$. Consider a new labeling φ' defined by

$$\begin{aligned}
&\varphi'(S) = \varphi^*(S) && \text{for all } S \in \mathcal{S} \text{ such that } 1 \leq \varphi^*(S) \leq \varphi^*(S_i)\\
&\varphi'(S) = \varphi^*(S) + 1 && \text{for all } S \in \mathcal{S} \text{ such that } \varphi^*(S_i) < \varphi^*(S) < \varphi^*(S_j)\\
&\varphi'(S_j) = \varphi^*(S_k) &&\\
&\varphi'(S) = \varphi^*(S) + 1 && \text{for all } S \in \mathcal{S} \text{ such that } \varphi^*(S_j) < \varphi^*(S) \leq m
\end{aligned}$$

For simplicity of notation, we write i instead of $\varphi^*(S_i)$, j instead of $\varphi^*(S_j)$ and k instead of $\varphi^*(S_k)$. Observe that $i + 1 = k$. We claim that φ' is an sbl for $\mathcal{S}$. It is sufficient to show that

$$\bigcup_{\varphi'(S) \leq \ell} S \subseteq \bigcup_{\varphi^*(S) \leq \ell} S$$

for all $k \leq \ell \leq j$. We check at once that

$$
\begin{aligned}
\bigcup_{1 \leq \varphi'(S) \leq \ell} S &= \left(\bigcup_{1 \leq \varphi'(S) \leq i} S \right) \cup S_j \cup \left(\bigcup_{k+1 \leq \varphi'(S) \leq \ell} S \right) \\
&= \left(\bigcup_{1 \leq \varphi^*(S) \leq i} S \right) \cup S_j \cup \left(\bigcup_{k \leq \varphi^*(S) \leq \ell-1} S \right) \\
&= \left(\bigcup_{1 \leq \varphi^*(S) \leq \ell-1} S \right) \cup S_j \\
&= \bigcup_{1 \leq \varphi^*(S) \leq \ell-1} S \qquad\qquad \text{(since } S_i = S_j\text{)} \\
&\subseteq \bigcup_{1 \leq \varphi^*(S) \leq \ell} S
\end{aligned}
$$

It follows immediately that $S_j \notin \mathcal{M}(\varphi')$. Indeed, $\varphi^*(S_j) \leq \varphi^*(S)$ for all $S \in \mathcal{M}(\varphi^*)$.

We proceed to show that $|\mathcal{M}(\varphi^*)| > |\mathcal{M}(\varphi')|$. If $|\mathcal{M}(\varphi')| = 0$, we are done, so that we may assume $|\mathcal{M}(\varphi')| > 0$. Let $S' \in \mathcal{M}(\varphi')$. Then there exist subsets S_a and S_b such that $\varphi'(S_a) < \varphi'(S_b) < \varphi'(S')$ and $S_a = S' \neq S_b$. Now, observe that we must have $\varphi^*(S_j) < \varphi^*(S')$. Again, this follows from the fact that $\varphi^*(S_j) \leq \varphi^*(S)$ for all $S \in \mathcal{M}(\varphi^*)$. Therefore, by construction of the labeling φ', either $\varphi^*(S_a) < \varphi^*(S_b) < \varphi^*(S')$ or $\varphi^*(S_b) < \varphi^*(S_a) < \varphi^*(S')$. We claim that $S' \in \mathcal{M}(\varphi^*)$. The result is certainly valid in case $\varphi^*(S_a) < \varphi^*(S_b)$. Suppose now that $\varphi^*(S_b) < \varphi^*(S_a)$. Then it follows that we must have $S_a = S_j$. But $S_i = S_j$. Therefore $\varphi^*(S_i) < \varphi^*(S_b) < \varphi^*(S')$ and $S_i = S' \neq S_b$, and hence $S' \in \mathcal{M}(\varphi^*)$. Combining this with the fact that $S_j \notin \mathcal{M}(\varphi')$ yields $|\mathcal{M}(\varphi^*)| > |\mathcal{M}(\varphi')|$. This contradicts the choice of φ^*. Therefore, we must have $|\mathcal{M}(\varphi^*)| = 0$ and hence φ^* is a normalized sbl of $\mathcal{S}$. □

Proof (of Lemma 3). We claim that there exists a satisfying truth assignment f for ϕ if and only if there exists an sbl of the configuration of subsets C of the ground set $\mathbf{X}$.

Suppose that there exists a satisfying truth assignment f for the 3CNF formula ϕ. Due to symmetry, there is no loss of generality in assuming that each clause is satisfied by its first literal. Define an labeling τ of the configuration of subsets C as follows. First, we begin our labeling construction by placing all the empty subsets of C, that is

$$1 \leq \tau(\mathcal{B}_i) \leq p_1 - 1$$

for all $\mathcal{B}_i \in B$. Next, all the subsets corresponding to the variables are ordered as follows: for all $\mathcal{V}_{i,j} \in \mathcal{V}_i$, $1 \leq i \leq n$,

$$p_i + \sum_{k=1}^{i-1}(p_k + q_k + r_k) \leq \tau(\mathcal{V}_{i,j}) \leq p_i + q_i + \sum_{k=1}^{i-1}(p_k + q_k + r_k) - 1$$

The satisfying truth assignment f for ϕ is coded in our construction as follows: for all $\mathcal{A}_{i,j} \in \mathcal{A}_i$, $1 \leq i \leq n$, where $\mathcal{A}_i = \mathcal{T}_i$ if $f(x_i) = \text{TRUE}$ and $\mathcal{A}_i = \mathcal{F}_i$ if $f(x_i) = \text{FALSE}$,

$$p_i + q_i + \sum_{k=1}^{i-1}(p_k + q_k + r_k) \leq \tau(\mathcal{A}_{i,j}) \leq \sum_{k=1}^{i}(p_k + q_k + r_k) - 1$$

and for all $\mathcal{A}'_{i,j} \in \mathcal{A}'_i$, $1 \leq i \leq n$, where $\mathcal{A}'_i = \mathcal{T}'_i$ if $f(x_i) = \text{TRUE}$ and $\mathcal{A}'_i = \mathcal{F}'_i$ if $f(x_i) = \text{FALSE}$,

$$\sum_{k=1}^{i}(p_k + q_k + r_k) \leq \tau(\mathcal{A}'_{i,j}) \leq p_{i+1} + \sum_{k=1}^{i}(p_k + q_k + r_k) - 1$$

It easily follows that

$$\tau(\mathcal{V}_i) < \tau(\mathcal{A}_i) < \tau(\mathcal{A}'_i) < \tau(\mathcal{V}_j) < \tau(\mathcal{A}_j) < \tau(\mathcal{A}'_j)$$

for all $1 \leq i < j \leq n$.

Having disposed of all those subsets corresponding to the variables and to the satisfying truth assignment f for ϕ, we now turn to the clauses. This will be divided into two parts. First, the subsets of $\mathcal{C}_i$, $1 \leq i \leq m$, are ordered as follows: for all $\mathcal{C}_{i,j} \in \mathcal{C}_i$, $1 \leq i \leq m$,

$$p_{n+i} + \sum_{k=1}^{n+i-1}(p_k + q_k + r_k) \leq \tau(\mathcal{C}_{i,j}) \leq p_{n+i} + q_{n+i} + \sum_{k=1}^{n+i-1}(p_k + q_k + r_k) - 1$$

Second, all the subsets of $\mathcal{L}_{i,1}$ and $\mathcal{L}'_{i,1}$, $1 \leq i \leq m$, are ordered as follows[1]: for all $\mathcal{L}_{i,1,j} \in \mathcal{L}_{i,1}$, $1 \leq i \leq m$,

$$p_{n+i} + q_{n+i} + \sum_{k=1}^{n+i-1}(p_k + q_k + r_k) \leq \tau(\mathcal{L}_{i,1,j}) \leq \sum_{k=1}^{n+i}(p_k + q_k + r_k) - 1$$

and for all $\mathcal{L}'_{i,1,j} \in \mathcal{L}'_{i,1}$, $1 \leq i \leq m$,

$$\sum_{k=1}^{n+i}(p_k + q_k + r_k) \leq \tau(\mathcal{L}'_{i,1,j}) \leq p_{n+i+1} + \sum_{k=1}^{n+i}(p_k + q_k + r_k) - 1$$

[1] Recall that we assume that each clause is satisfied by its first literal.

A trivial verification shows that

$$\tau(\mathcal{C}_i) < \tau(\mathcal{L}_{i,1}) < \tau(\mathcal{L}'_{i,1}) < \tau(\mathcal{C}_j) < \tau(\mathcal{L}_{j,1}) < \tau(\mathcal{L}'_{j,1})$$

for all $1 \leq i < j \leq m$.

Here come the K separator subsets of $\mathcal{S}$, that is for all $\mathcal{S}_i \in \mathcal{S}$

$$p_{n+m+1} + \sum_{k=1}^{n+m} (p_k + q_k + r_k) \leq \tau(\mathcal{S}_i) \leq p_{n+m+1} + K + \sum_{k=1}^{n+m} (p_k + q_k + r_k) - 1$$

Up to now, all the subsets of C but L of them have been used to define τ. Our labeling construction ends with those subsets by packing them in the L last places:

$$p_{n+m+1}+K+\sum_{k=1}^{n+m}(p_k+q_k+r_k) \leq \tau(\mathcal{X}_j) \leq p_{n+m+1}+K+L+\sum_{k=1}^{n+m}(p_k+q_k+r_k)-1$$

A careful examination of τ shows that $\left|\bigcup_{\tau(X)\leq i} X\right| \leq i$ for $1 \leq i \leq |C|$ and hence that τ is an sbl[2] of the configuration of subsets C of the ground set $\mathbf{X}$. An (partial) illustration of the construction of the sbl τ is shown in Figure 1.

For the converse, suppose that there exists an sbl τ for C, that is an labeling such that $\left|\bigcup_{\tau(X_j)\leq i} X_j\right| \leq i$ for $1 \leq i \leq |C|$. According to lemma 2, there is no loss of generality in assuming that τ is a normalized sbl *i.e.*, the identical subsets of the configuration C are mapped by τ to a set of consecutive integers. This property is crucial in our proof.

Let us start by proving that $1 \leq \tau(\mathcal{B}_i) \leq p_1 - 1$ for all $\mathcal{B}_i \in \mathcal{B}$. Indeed, as mentioned earlier in the proof, all the above defined subsets of the configuration of subsets C but those of $\mathcal{B}$ contain at least p_1 elements and hence can not be mapped to an integer $i \leq p_1 - 1$ in any sbl τ for C.

Define the subcollection $C^* \subset C$ as follows:

$$C^* = (X \in C \mid \max\{\tau(\mathcal{B}_i) \mid \mathcal{B}_i \in \mathcal{B}\} < \tau(X) < \min\{\tau(\mathcal{S}_i) \mid \mathcal{S}_i \in \mathcal{S}\})$$

In other words, the subcollection C^* contains all those subsets that go after the last bootstrap subset $\mathcal{B}_i \in \mathcal{B}$ and before the first separator subset $\mathcal{S}_i \in \mathcal{S}$. We claim that C^* contains (1) the subsets of $\mathcal{V}_i$ for $1 \leq i \leq n$, (2) either the subsets of $\mathcal{T}_i$ and $\mathcal{T}'_i$ or the subsets of $\mathcal{F}_i$ and $\mathcal{F}'_i$ for $1 \leq i \leq n$, (3) the subsets of $\mathcal{C}_i$ for $1 \leq i \leq m$ and (4) the subsets of $\mathcal{L}_{i,1}$ and $\mathcal{L}'_{i,1}$ or the subsets of $\mathcal{L}_{i,2}$ and $\mathcal{L}'_{i,2}$ or the subsets of $\mathcal{L}_{i,3}$ and $\mathcal{L}'_{i,3}$ for $1 \leq i \leq m$. For simplicity, this will be divided into several steps.

Having disposed of the bootstrap subsets, we now turn to the non-empty subsets of the configuration of subsets C. Roughly speaking, we show that our

[2] Observe that the construction of τ may be specialized to yield a normalized sbl. Indeed, all the subsets of C but the L last ones are mapped by τ to a set of consecutive integers. But there is no less of generality in assuming that these subsets are mapped by τ to a set of consecutive integers as well.

construction implies a *force placement* of the subsets of C together with choices corresponding to a satisfying truth assignment for the 3CNF formula ϕ. First, we must have:

$$p_1 \leq \tau(\mathcal{V}_{1,j}) \leq p_1 + q_1 - 1$$

for all $1 \leq j \leq q_1$. Indeed, all other subsets have more than p_1 elements and hence can not be mapped to p_1 by τ. Moreover, τ is a normalized sbl for C, *i.e.*, the subsets $\mathcal{V}_{1,j}$, $1 \leq j \leq q_1$, are mapped by τ to a set of consecutive integers. Now, the key point is that the subset which is mapped to $p_1 + q_1$ by τ must have cardinality at most $p_1 + q_1$. But a careful examination of the configuration of subsets C shows that all the remaining subsets contain $\mathbf{V}_1$, and hence the subset which is mapped to $p_1 + q_1$ by τ may introduce at most q_1 new elements of the ground set. Then it follows from our construction that this subset is either a subset of the collection $\mathcal{T}_1$ or a subset of the collection $\mathcal{F}_1$. Indeed, it is sufficient to note that $r_1 + p_2 > q_1$, and hence that no subset $\mathcal{V}_{2,j} \in \mathcal{V}_2$ can satisfy $\tau(\mathcal{V}_{2,j}) = p_1 + q_1$. As an immediate result, exactly one of the following two statements is true:

$$p_1 + q_1 \leq \tau(\mathcal{T}_{1,j}) \leq p_1 + q_1 + r_1 - 1 \qquad (1)$$

$$p_1 + q_1 \leq \tau(\mathcal{F}_{1,j}) \leq p_1 + q_1 + r_1 - 1 \qquad (2)$$

for all $1 \leq j \leq r_1$. We can now proceed analogously to the above to obtain:

$$(1) \quad \Rightarrow \quad p_1 + q_1 + r_1 \leq \tau(\mathcal{T}'_{1,j}) \leq p_1 + q_1 + r_1 + p_2 - 1$$

$$(2) \quad \Rightarrow \quad p_1 + q_1 + r_1 \leq \tau(\mathcal{F}'_{1,j}) \leq p_1 + q_1 + r_1 + p_2 - 1$$

for all $1 \leq j \leq p_2$. Summarizing, exactly one of the following two statements is true:

$$p_1 \leq \tau(\mathcal{V}_1) < \tau(\mathcal{T}_1) < \tau(\mathcal{T}'_1) \leq p_1 + q_1 + r_1 + p_2 - 1$$

$$p_1 \leq \tau(\mathcal{V}_1) < \tau(\mathcal{F}_1) < \tau(\mathcal{F}'_1) \leq p_1 + q_1 + r_1 + p_2 - 1$$

We continue in this fashion obtaining that exactly one of the following two statements is true:

$$p_i + \sum_{k=1}^{i-1}(p_k + q_k + r_k) \leq \tau(\mathcal{V}_i) < \tau(\mathcal{T}_i) < \tau(\mathcal{T}'_i) \leq p_{i+1} + \sum_{k=1}^{i}(p_k + q_k + r_k) - 1$$

$$p_i + \sum_{k=1}^{i-1}(p_k + q_k + r_k) \leq \tau(\mathcal{V}_i) < \tau(\mathcal{F}_i) < \tau(\mathcal{F}'_i) \leq p_{i+1} + \sum_{k=1}^{i}(p_k + q_k + r_k) - 1$$

for all $1 \leq i \leq n$. This result is crucial as it allows us to construct a truth assignment f for the 3CNF formula ϕ. Indeed, subsets of $\mathcal{T}_i$ and $\mathcal{T}'_i$ are associated in our construction to literal x_i while subsets of $\mathcal{F}_i$ and $\mathcal{F}'_i$ are associated in our construction to literal $\overline{x}_i$.

Having disposed of the variables we now turn to the clauses. For the sake of clarity, let us first introduce the temporary notations

$$l_i = p_{n+i} + \sum_{k=1}^{n+i-1} (p_k + q_k + r_k)$$
$$h_i = p_{n+i+1} + \sum_{k=1}^{n+i} (p_k + q_k + r_k) - 1$$

for $1 \leq i \leq m$. We may now proceed analogously to the above to obtain that exactly one of the following three statements is true:

$$l_i \leq \tau(\mathcal{C}_i) < \tau(\mathcal{L}_{i,1}) < \tau(\mathcal{L}'_{i,1}) \leq h_i$$
$$l_i \leq \tau(\mathcal{C}_i) < \tau(\mathcal{L}_{i,2}) < \tau(\mathcal{L}'_{i,2}) \leq h_i$$
$$l_i \leq \tau(\mathcal{C}_i) < \tau(\mathcal{L}_{i,3}) < \tau(\mathcal{L}'_{i,3}) \leq h_i$$

for all $1 \leq i \leq m$.

According to the above, the subcollection C^* contains either the subsets of $\mathcal{T}_i$ and $\mathcal{T}'_i$ or the subsets of $\mathcal{F}_i$ and $\mathcal{F}'_i$ for $1 \leq i \leq n$. Therefore we can define a truth assignment f for the 3CNF formula ϕ as follows: $f(x_i) = \text{TRUE}$ if $\mathcal{T}_i \subset C^*$ and $f(x_i) = \text{FALSE}$ if $\mathcal{F}_i \subset C^*$ for $1 \leq i \leq n$. We claim that f is a satisfying truth assignment for ϕ. Indeed, for each i, $1 \leq i \leq m$, consider the true statement

$$l_i \leq \tau(\mathcal{C}_i) < \tau(\mathcal{L}_{i,k}) < \tau(\mathcal{L}'_{i,k}) \leq h_i$$

where k is either 1, 2 or 3. By construction we have

$$\mathcal{L}_{i,k,j} = \mathbf{V} \cup \mathbf{V}' \cup \mathbf{A}_k \cup \mathbf{L}_{i,k} \cup \left(\bigcup_{1 \leq \ell \leq i} \mathbf{C}_\ell \right) \cup \left(\bigcup_{1 \leq \ell \leq i-1} \mathbf{C}'_\ell \right)$$

for $1 \leq j \leq r_{n+i}$ and

$$\mathcal{L}'_{i,k,j} = \mathbf{V} \cup \mathbf{V}' \cup \mathbf{A}_k \cup \mathbf{L}_{i,k} \cup \left(\bigcup_{1 \leq \ell \leq i} \mathbf{C}_\ell \right) \cup \left(\bigcup_{1 \leq \ell \leq i} \mathbf{C}'_\ell \right)$$

for $1 \leq j \leq p_{n+i+1}$, where $\mathbf{A}_k = \mathbf{T}_\ell$ if $\lambda_{i,k} = x_\ell$ and $\mathbf{A}_k = \mathbf{F}_\ell$ if $\lambda_{i,k} = \overline{x}_\ell$. But a careful examination of the configuration of subsets C and of the sbl τ shows that none of the previous subsets contain the elements of $\mathbf{L}_{i,k}$. Since $|\mathcal{C}_i| = q_{n+i}$ and $|\mathbf{L}_{i,k}| = q_{n+i}$, then it follows that all the elements of $\mathbf{A}_k$ must have been introduced by previous subsets. Indeed, $l_i \leq \tau(\mathcal{C}_i) < \tau(\mathcal{L}_{i,k}) < \tau(\mathcal{L}'_{i,k}) \leq h_i$ and hence $\mathcal{C}_i$ and $\mathcal{L}_{i,k}$ are mapped by τ to a set of consecutive integers. Therefore, if $\mathbf{A}_k = \mathbf{T}_\ell$ (resp. $\mathbf{A}_k = \mathbf{F}_\ell$) then we must have chosen $\mathcal{T}_\ell$ (resp. $\mathcal{F}_\ell$) in a previous step, that is $\mathcal{T}_\ell, \mathcal{T}'_\ell \in C^*$ (resp. $\mathcal{F}_\ell, \mathcal{F}'_\ell \in C^*$). Then it follows that clause c_i is satisfied by its k-th literal. Hence, f is a satisfying truth assignment for ϕ and the lemma is proved. □

Proof (of Lemma 4). The proof is by induction. The result is obvious for $n = 1$. For the inductive step, assume the statement holds for some natural number n and let A be a $(0,1)$-matrix of order $n+1$. Since A is assumed to be a pet matrix there exist permutation matrices P and Q such that $PAQ \leq \triangle_n$. Let $A' = PAQ$. As $\text{per}(A)$ remains invariant under arbitrary permutation of the lines of A we have $\text{per}(A) = \text{per}(A')$. If $\text{per}(A') = 0$, we are done. Otherwise $\text{per}(A') \geq 1$ and hence n 1's appear on the main diagonal of A'. We designate the matrix obtained from A' by deleting the first row and the first column permuted matrix by A''. As the first row of A' contains exactly one 1, we have $\text{per}(A'') = \text{per}(A') = \text{per}(A) \leq 1$, thereby proving the lemma. □

Proof (of Lemma 5). The proof is by induction. The assertion is certainly valid for $n = 2$ as all $(0,1)$-matrix of order 2 are pet matrices but J_2. For the inductive step, assume the statement holds for some natural number n and let A be $(0,1)$-matrix of order $n+1$. If $A = \mathbf{0}$ we are done. Otherwise, let α_i be a row of A with minimum row sum r_i. Since A contains at most $n+1$ 1's, we certainly have $r_i \leq 1$. We need to consider two cases:

- If $r_i = 0$, let β_j be a column of A with maximum column sum c_j. Since $A \neq \mathbf{0}$, we certainly have $c_j > 0$. We now permute the lines of A so that row α_i and column β_j are the first row and column of the permuted matrix. We now delete the first row and column of this permuted matrix and apply the induction hypothesis to this submatrix of order n.
- If $r_i = 1$, we may permute the rows of A so that row 1 of the permuted matrix contains a 1 in the $(1,1)$ position and O's elsewhere. We now delete the first row and column of this matrix and apply the induction hypothesis to this submatrix of order n.

□

Proof (of Lemma 7). The proof is by induction on m. First, the assertion is certainly valid for $m = 1$ as

$$\begin{aligned} 2\,\alpha(2^1)\,(T(2^0)+1) &= 2 \times 4 \times (1+1) \\ &= 16 \end{aligned}$$

and

$$\begin{aligned} \left(2^1 \prod_{i=1}^{1} \alpha(2^i)\right) + \left(\sum_{i=1}^{1} 2^{1-i+1} \prod_{j=i}^{1} \alpha(2^j)\right) &= (2^1 \times \alpha(2^1)) + (2^1 \times \alpha(2^1)) \\ &= (2 \times 4) + (2 \times 4) \\ &= 16. \end{aligned}$$

Suppose now that the assertion is true for 2^m. Then

$$
\begin{aligned}
T(2^{m+1}) &= 2\,\alpha(2^{m+1})\,T(2^m) + 2\,\alpha(2^{m+1}) \\
&= 2\,\alpha(2^{m+1}) \left(\left(2^m \prod_{i=1}^{m} \alpha(2^i) \right) + \left(\sum_{i=1}^{m} 2^{m-i+1} \prod_{j=i}^{m} \alpha(2^j) \right) \right) + 2\,\alpha(2^{m+1}) \\
&= \left(2^{m+1} \prod_{i=1}^{m+1} \alpha(2^i) \right) + \left(\sum_{i=1}^{m} 2^{m+1-i+1} \prod_{j=i}^{m+1} \alpha(2^j) \right) + 2\,\alpha(2^{m+1}) \\
&= \left(2^{m+1} \prod_{i=1}^{m+1} \alpha(2^i) \right) + \left(\sum_{i=1}^{m+1} 2^{m+1-i+1} \prod_{j=i}^{m+1} \alpha(2^j) \right) - 2\,\alpha(2^{m+1}) + 2\,\alpha(2^{m+1}) \\
&= \left(2^{m+1} \prod_{i=1}^{m+1} \alpha(2^i) \right) + \left(\sum_{i=1}^{m+1} 2^{m+1-i+1} \prod_{j=i}^{m+1} \alpha(2^j) \right).
\end{aligned}
$$

□

Proof (of Lemma 8). We need the following claims.

Claim 1.

$$\alpha(2^i) = \frac{2^{2^{m+1}+m-1}}{\pi} \left(1 + O\left(2^{1-m}\right)\right).$$

Proof. We use the following well-known asymptotic (see e.g. [5]):

$$\binom{2n}{n} = \frac{4^n}{\sqrt{\pi n}} \left(1 + O\left(n^{-1}\right)\right).$$

Therefore,

$$
\begin{aligned}
\alpha(2^m) &= 2^{2m-2} \binom{2^m}{2^{m-1}}^2 \\
&= 2^{2m-2} \left(\frac{4^{2^{m-1}}}{\sqrt{\pi 2^{m-1}}} \left(1 + O\left(2^{1-m}\right)\right) \right)^2 \\
&= 2^{2m-2} \left(\frac{4^{2^{m-1}}}{\sqrt{\pi 2^{m-1}}} \right)^2 \left(1 + O\left(2^{1-m}\right)\right)^2 \\
&= 2^{2m-2} \frac{4^{2^m}}{\pi 2^{m-1}} \left(1 + O\left(2^{1-m}\right)\right)^2 \\
&= \frac{2^{2^{m+1}+m-1}}{\pi} \left(1 + O\left(2^{1-m}\right)\right)^2.
\end{aligned}
$$

We now develop the error term to obtain

$$\begin{aligned}\alpha(2^m) &= \frac{2^{2^{m+1}+m-1}}{\pi}\left(1+O\left(2^{1-m}\right)\right)\left(1+O\left(2^{1-m}\right)\right)\\ &= \frac{2^{2^{m+1}+m-1}}{\pi}\left(1+O\left(2^{1-m}\right)+O\left(2^{1-m}\right)+O\left(2^{2-2m}\right)\right)\\ &= \frac{2^{2^{m+1}+m-1}}{\pi}\left(1+O\left(2^{1-m}\right)\right).\end{aligned}$$

□

Claim 2.

$$\prod_{i=1}^{m}\alpha(2^i) = O\left(\frac{2^{2^{m+2}-3}}{\pi^m}\right).$$

Proof. According to Claim 1, we may write

$$\begin{aligned}\prod_{i=1}^{m}\alpha(2^i) &= \prod_{i=1}^{m}\frac{2^{2^{i+1}+i-1}}{\pi}\left(1+O\left(2^{1-i}\right)\right)\\ &= \left(\frac{1}{2\pi}\right)^m\left(\prod_{i=1}^{m}2^{2^{i+1}}\right)\left(\prod_{i=1}^{m}2^{i}\right)\left(\prod_{i=1}^{m}\left(1+O\left(2^{1-i}\right)\right)\right)\\ &= \left(\frac{1}{2\pi}\right)^m\left(2^{2^{m+2}-4}\right)\left(2^{m+1}-2\right)\left(\prod_{i=1}^{m}\left(1+O\left(2^{1-i}\right)\right)\right)\\ &= \frac{2^{2^{m+2}-3}}{\pi^m}\left(\frac{2^m-1}{2^m}\right)\left(\prod_{i=1}^{m}\left(1+O\left(2^{1-i}\right)\right)\right)\\ &= \frac{2^{2^{m+2}-3}}{\pi^m}\left(1-2^{-m}\right)\left(\prod_{i=1}^{m}\left(1+O\left(2^{1-i}\right)\right)\right).\end{aligned}$$

But $\prod_{i=1}^{m}\left(1+O\left(2^{1-i}\right)\right) = O(1)$ and hence

$$\begin{aligned}\prod_{i=1}^{m}\alpha(2^i) &= \frac{2^{2^{m+2}-3}}{\pi^m}O\left(1-2^{-m}\right)\\ &= O\left(\frac{2^{2^{m+2}-3}}{\pi^m}\right).\end{aligned}$$

□

Having disposed of the preliminary steps, write $\beta(i) = 2^{m-i}\prod_{j=i}^{m}\alpha(2^j)$ so that our goal reduces to evaluating $\sum_{i=1}^{m}\beta(i)$. For $1 \leq i < m$, we first observe that

$$\begin{aligned}
2\,\beta(i+1) &= 2\;2^{m-i-1}\prod_{j=i+1}^{m}\alpha(2^j)\\
&= 2^{m-i}\prod_{j=i+1}^{m}\alpha(2^j)\\
&\le 2^{m-i}\prod_{j=i}^{m}\alpha(2^j) \qquad (\text{since } \alpha(i)\ge 1)\\
&= \beta(i).
\end{aligned}$$

Therefore

$$\begin{aligned}
\sum_{i=1}^{m}\beta(i) &\le \sum_{i=1}^{m}\frac{\beta(1)}{2^{i-1}}\\
&= 2\,\beta(1)\sum_{i=1}^{m}2^{-i}\\
&= 2\beta(1)\left(1-2^{-m}\right)\\
&\le 2\beta(1).
\end{aligned}$$

Then it follows that

$$\begin{aligned}
\sum_{i=1}^{m}2^{m-i}\prod_{j=i}^{m}\alpha(2^j) &\le 2\,\beta(1)\\
&= 2\;2^{m-1}\prod_{j=1}^{m}\alpha(2^j)\\
&= 2^m\,O\left(\frac{2^{2^{m+2}-3}}{\pi^m}\right)\\
&= O\left(\frac{2^{2^{m+2}+m-3}}{\pi^m}\right).
\end{aligned}$$

□