Discrete Applied Mathematics 193 (2015) 11–38

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Synthesizing cubes to satisfy a given intersection pattern

Weikang Qian [a], Marc D. Riedel [b,*], Ivo Rosenberg [c]

[a] University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA
[c] Mathematics and Statistics, University of Montreal, Montreal, Quebec, Canada H3C 3J7

ARTICLE INFO

Article history:
Received 5 June 2011
Received in revised form 20 March 2015
Accepted 25 March 2015
Available online 11 May 2015

Keywords:
Boolean product
Cube
Minterm
Two-level logic synthesis

ABSTRACT

In two-level logic synthesis, the typical input specification is a set of minterms defining the *on* set and a set of minterms defining the *don't care* set of a Boolean function. The problem is to synthesize an optimal set of product terms, or cubes, that covers all the minterms in the *on* set and some of the minterms in the *don't care* set. In this paper, we consider a different specification: instead of the *on* set and the *don't care* set, we are given a set of numbers, each of which specifies the number of minterms covered by the intersection of one of the subsets of a set of λ cubes. We refer to the given set of numbers as an *intersection pattern*. The problem is to determine whether there exists a set of λ cubes that satisfies the given intersection pattern and, if it exists, to synthesize the set of cubes. We show a necessary and sufficient condition for the existence of λ cubes to satisfy a given intersection pattern. We also show that the synthesis problem can be reduced to the problem of finding a non-negative solution to a set of linear equations and inequalities.

1. Introduction

Two-level logic synthesis is a well-developed and mature topic [4,7]. The typical input specification for a two-level synthesis problem is the *on* set and the *don't care* set (or in some cases, the *off* set) of a Boolean function. The *on* set and the *don't care* set consist of minterms that define when the function evaluates to one and when its evaluation can be either zero or one, respectively. The problem is to synthesize an optimal set of product terms, or *cubes*, that covers all the minterms in the *on* set and some of the minterms in the *don't care* set.

In this work, we consider a related yet different problem pertaining to the synthesis of a set of cubes. A set of cubes, besides defining a Boolean function, also defines a set of numbers, each of which corresponds to the number of minterms covered by the intersection of one of the subsets of the set of cubes. For example, given a set of three cubes on four variables x_0, x_1, x_2, x_3, which are $c_0 = x_0x_1$, $c_1 = x_2$, and $c_2 = x_1x_3$, the numbers of minterms covered by $c_0, c_1, c_2, c_0c_1, c_0c_2, c_1c_2$, and $c_0c_1c_2$ are 4, 8, 4, 2, 2, 2, and 1, respectively. We refer to this set of numbers as an *intersection pattern*.

Given a set of λ cubes, it is trivial to get its intersection pattern, which is a set of $2^\lambda - 1$ numbers. However, it is nontrivial to answer the reverse problem: given a set of $2^\lambda - 1$ numbers, can we obtain a set of λ cubes so that its intersection pattern equals the given set of numbers, or prove that there does not exist such a set of λ cubes? We call this problem the *λ-cube intersection problem*. It is what we intend to solve in this paper.

In this paper, we will deal with the number of minterms contained by a Boolean function. For simplicity, we use the following definition:

* Corresponding author.
E-mail addresses: qianwk@sjtu.edu.cn (W. Qian), mriedel@umn.edu (M.D. Riedel), rosenb@dms.umontreal.ca (I. Rosenberg).

http://dx.doi.org/10.1016/j.dam.2015.03.012
0166-218X/

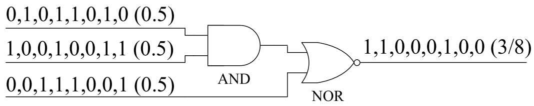

Fig. 1. An AND gate followed by a NOR gate transforms three independent random inputs of probability 0.5 of being one into an random output of probability $\frac{3}{8}$ of being one. The inputs and output of the circuit are random bit streams. The numbers in the parentheses denote the probabilities.

Definition 1. Define $V(f)$ to be the number of minterms contained in a Boolean function f. □

The following example shows an instance of the λ-cube intersection problem.

Example 1. In a 3-cube intersection problem on 4 variables x_0, x_1, x_2, x_3, suppose that we require the intersection pattern to be

$$V(c_0) = 4, \quad V(c_1) = 8, \quad V(c_2) = 4,$$
$$V(c_0c_1) = V(c_0c_2) = V(c_1c_2) = 2, \quad V(c_0c_1c_2) = 1.$$

We can synthesize cubes $c_0 = x_0x_1$, $c_1 = x_2$, and $c_2 = x_1x_3$ to satisfy that pattern. □

The motivation of our study of the λ-cube intersection problem is that it pertains to synthesizing logic circuits for probabilistic computation, a new paradigm that we have advocated [6]. A fundamental problem in this context is the so called *arithmetic two-level minimization problem*. In the remaining part of the introduction, we will introduce this problem and outline our proposed solution to it. As we will show, an important step in our solution to the arithmetic two-level minimization problem is to solve the λ-cube intersection problem. This motivates our study of the λ-cube intersection problem in this work.

1.1. Arithmetic two-level minimization problem

In the paradigm of probabilistic logical computation, digital circuits are designed to transform a set of input probabilities, encoded by random bit streams, into output probabilities, also encoded by random bit streams [6]. A fundamental problem in this context is how to synthesize combinational logic that takes independent inputs with probability 0.5 of being one and generates other probabilities as outputs. For example, we can use the combinational circuit shown in Fig. 1 to generate an output probability $\frac{3}{8}$ from three independent input probabilities 0.5.

For a combinational circuit with n inputs, if each input has probability 0.5 of being one and all the inputs are independent, then each input combination has probability of $\frac{1}{2^n}$ of occurring. If the Boolean function contains exactly m minterms, then the probability that the output is one is $\frac{m}{2^n}$. Conversely, if we want to synthesize a probability $\frac{m}{2^n}(0 \le m \le 2^n)$, we can simply implement it with a Boolean function of m minterms. However, there are $\binom{2^n}{m}$ Boolean functions that contain exactly m minterms and different functions have different implementation cost. This motivates a new problem in logic synthesis: if we want to synthesize a logic circuit such that it covers exactly m minterms, while which m minterms are covered does not matter, then how can we design an optimal logic circuit?

We focus on two-level implementation of logic circuits [4]. Minimizing the area of the two-level implementation is equivalent to minimizing the number of cubes of the sum-of-product (SOP) representation of a Boolean function [4]. Thus, the problem, which we will refer to as the *arithmetic two-level minimization problem*, can be formulated as:

Given the number of variables n for a Boolean function and an integer $0 \le m \le 2^n$, *find an SOP Boolean expression with the minimum number of cubes that contains exactly m minterms.*

Given m and n, there exists a simple procedure to synthesize a small number of cubes to cover exactly m minterms [5]. The number of cubes synthesized by this procedure is equal to the number of ones in the binary representation of m. Suppose that the binary representation of m has k ones and $m = \sum_{i=0}^{k-1} 2^{m_i}$, where $m_0 < m_1 < \cdots < m_{k-1}$. Then we can easily find k cubes $c_0, c_1, \ldots, c_{k-1}$ so that (1) c_i contains 2^{m_i} minterms, and (2) any two different cubes c_i and c_j are disjoint, i.e., $c_i \cdot c_j = 0$. These k cubes together cover exactly m minterms.

For example, consider $m = 7$ and $n = 4$. The binary representation of m has 3 ones and $m = 2^2 + 2^1 + 2^0$. We can construct three cubes $c_0 = x_0\bar{x}_1\bar{x}_2\bar{x}_3$, $c_1 = x_1\bar{x}_2\bar{x}_3$, and $c_2 = x_2\bar{x}_3$ to cover 7 minterms. Note that cubes c_0, c_1, c_2 cover 1, 2, 4 minterms, respectively. Further, they are mutually disjoint. Therefore, the total number of minterms covered by these three cubes is 7.

However, the above method cannot guarantee to give the *minimum* number of cubes to cover m minterms. For the above example, indeed, we can cover 7 minterms with two cubes: $c_0 = x_0x_1$ and $c_1 = x_2x_3$. Note that both c_0 and c_1 contain 4 minterms; their intersection $c_0c_1 = x_0x_1x_2x_3$ contains one minterm. Thus, the total number of minterms covered by c_0 and c_1 is 7. Therefore, the above method only gives an upper bound on the minimum number of cubes for the arithmetic two-level minimization problem. Thus, a more sophisticated method is required.

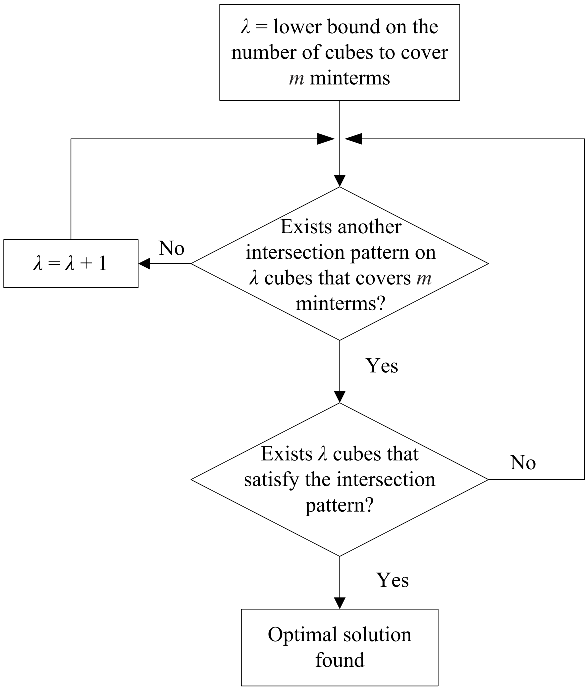

Fig. 2. The flow of our proposed search-based approach for solving the arithmetic two-level minimization problem. The λ-cube intersection problem is an important subproblem in the flow.

1.2. The relation between λ-cube intersection problem and arithmetic two-level minimization problem

In this section, we will outline our proposed solution to the arithmetic two-level minimization problem. As we will show, our solution hinges on solving the λ-cube intersection problem.

The simple procedure stated above cannot guarantee to give an optimal solution to an arithmetic two-level minimization problem. We propose a search-based approach to find the optimal solution. Studying the previous example, we find that an optimal solution potentially involves a set of non-disjoint cubes. Therefore, in our approach, a crucial subroutine is to count the number of minterms covered by a set of non-disjoint cubes. We apply the *inclusion–exclusion principle* for this purpose:

Given λ *cubes* $c_0, \ldots, c_{\lambda-1}$, *the number of minterms covered by the union of the* λ *cubes is*

$$V\left(\bigvee_{i=0}^{\lambda-1} c_i\right) = \sum_{i=0}^{\lambda-1} V(c_i) - \sum_{\substack{i,j:\\ 0\le i<j\le\lambda-1}} V(c_ic_j) + \sum_{\substack{i,j,k:\\ 0\le i<j<k\le\lambda-1}} V(c_ic_jc_k) - \cdots + (-1)^{\lambda-1}V\left(\prod_{i=0}^{\lambda-1} c_i\right). \tag{1}$$

The flow of our proposed search-based approach is shown in Fig. 2. We initially set λ to be a lower bound on the number of cubes to cover m minterms [5]. Then we will check whether we can find λ cubes so that they cover m minterms. In order to do so, we first construct an intersection pattern on λ cubes that covers m minterms, i.e., a set of $2^\lambda - 1$ numbers that let Eq. (1) evaluate to the target value m. Then, we need to check whether we can find λ cubes to satisfy that intersection pattern. If we find a solution to that instance of the λ-cube intersection problem, we obtain an optimal solution to the arithmetic two-level minimization problem. If not, we will try another intersection pattern on λ cubes. After all the intersection patterns on λ cubes that cover m minterms have been tried and no solution is found, we will increase λ by one. It can be seen that the λ-cube intersection problem is an important and recurring subproblem we will encounter in solving the arithmetic two-level minimization problem.

The following shows an example of applying our approach to solve an arithmetic two-level minimization problem.

Example 2. Synthesize an optimal SOP Boolean expression on 4 variables to cover 11 minterms.

Since we cannot cover 11 minterms with just 1 cube, the lower bound on the number of cubes is 2. Thus, initially, we set $\lambda = 2$. For $\lambda = 2$, we first construct an intersection pattern $\{V(c_0), V(c_1), V(c_0c_1)\}$, so that

$$V(c_0) + V(c_1) - V(c_0c_1) = 11.$$

One intersection pattern that satisfies the above equation is $V(c_0) = 8$, $V(c_1) = 4$ and $V(c_0c_1) = 1$. However, this 2-cube intersection problem has no solution. Thus, we will try another intersection pattern on 2 cubes that covers 11 minterms. Indeed, there is no other proper intersection pattern on 2 cubes that covers 11 minterms. Then, we raise λ to 3.

For $\lambda = 3$, we first construct an intersection pattern

$$\{V(c_0), V(c_1), V(c_2), V(c_0c_1), V(c_0c_2), V(c_1c_2), V(c_0c_1c_2)\},$$

so that

$$V(c_0) + V(c_1) + V(c_2) - V(c_0c_1) - V(c_0c_2) - V(c_1c_2) + V(c_0c_1c_2) = 11.$$

One intersection pattern that satisfies the above equation is $V(c_0) = 8$, $V(c_1) = 2$, $V(c_2) = 1$, and $V(c_0c_1) = V(c_0c_2) = V(c_1c_2) = V(c_0c_1c_2) = 0$. For that 3-cube intersection problem, we could synthesize cubes $c_0 = x_0$, $c_1 = \bar{x}_0x_1x_2$ and $c_2 = \bar{x}_0\bar{x}_1\bar{x}_2x_3$ to satisfy the given intersection pattern. Thus, we get an optimal solution of 3 cubes to the original arithmetic two-level minimization problem. □

In summary, in order to solve the arithmetic two-level minimization problem, it is critical to first solve the λ-cube intersection problem. In this work, we will focus on the λ-cube intersection problem.

The rest of the paper is organized as follows. In Section 2, we will introduce some preliminaries. In Section 3, we will give the solution to the λ-cube intersection problem of a special case. In Section 4, we will solve the general-case problem. In Section 5, we will discuss the implementation of the procedure to solve the λ-cube intersection problem. In Section 6, we show the performance of our solution on a number of test cases. We conclude the paper in Section 7.

2. Preliminaries

In this section, we will first introduce some basic definitions and then give a formal definition of the λ-cube intersection problem. Some of the basic definitions are adopted from [3].

The *n variables* of a Boolean function are denoted by $x_0, \ldots, x_{n-1}$. For a variable x, x and $\bar{x}$ are referred to as *literals*. A *Boolean product*, or product for short, is a conjunction of literals such that x and $\bar{x}$ do not appear simultaneously. For example, $x_1\bar{x}_2\bar{x}_3$ is a Boolean product. A Boolean product is also known as a *cube*, which is denoted by c. A *minterm* is a cube in which each of the n variables appear exactly once, in either its complemented or uncomplemented form. If cube c_2 takes the value one whenever cube c_1 equals one, we say that cube c_1 *implies* cube c_2 and write as $c_1 \subseteq c_2$. If cube c_1 implies cube c_2, then the number of minterms contained in cube c_1 is no larger than the number of minterms contained in cube c_2, i.e., $V(c_1) \le V(c_2)$. If $c_1 \cdot c_2 = 0$, we say that cubes c_1 and c_2 are *disjoint*.

If a cube c contains k literals ($0 \le k \le n$), then the number of minterms contained in the cube is $V(c) = 2^{n-k}$. Note that when a cube contains 0 literals, it is a special cube $c = 1$, which contains all the minterms in the entire Boolean space of n variables. There is another special cube called *empty cube*, which is $c = 0$. The number of minterms contained in the empty cube is $V(c) = 0$. Thus, the number of minterms contained in a cube is in the set $S = \{s|s = 0 \text{ or } s = 2^k, k = 0, 1, \ldots, n\}$.

In this paper, we deal with the intersection of cubes. To make the representation compact, we use the following definition.

Definition 2. Given a cube c and a $\gamma \in \{0, 1\}$, define

$$c^\gamma = \begin{cases} 1, & \text{if } \gamma = 0 \\ c, & \text{if } \gamma = 1. \end{cases}$$

Given a set of λ cubes $c_0, \ldots, c_{\lambda-1}$ and an integer $\Gamma = \sum_{i=0}^{\lambda-1} \gamma_i 2^i$, where $\gamma_i \in \{0, 1\}$, define C^Γ to be the intersection of a subset of cubes c_i's for those i's such that $\gamma_i = 1$, i.e., $C^\Gamma = \prod_{i=0}^{\lambda-1} c_i^{\gamma_i}$. □

For example, given three cubes c_0, c_1, and c_2, we have $C^5 = c_0c_2$. With the above definition, we can formally define the λ-cube intersection problem as follows:

Given $n > 0$, $\lambda > 0$, and a vector of 2^λ numbers $(v_0, v_1, \ldots, v_{2^\lambda-1})$, determine whether there exists a set of λ cubes $c_0, \ldots, c_{\lambda-1}$ on n variables $x_0, \ldots, x_{n-1}$, such that for all $0 \le \Gamma \le 2^\lambda - 1$, $V(C^\Gamma) = v_\Gamma$.

We refer to the vector of numbers $(v_0, \ldots, v_{2^\lambda-1})$ as *an intersection pattern* on λ cubes, or simply as an intersection pattern. If a set of λ cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the property that for any $0 \le \Gamma \le 2^\lambda - 1$, $V(C^\Gamma) = v_\Gamma$, then we say that the set of cubes satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$.

If there exists a set of λ cubes that satisfies the intersection pattern, then for all $0 \le \Gamma \le 2^\lambda - 1$, we have

$$v_\Gamma = V(C^\Gamma) \in S = \{s|s = 0 \text{ or } s = 2^k, k = 0, 1, \ldots, n\}.$$

Further, the number $v_0 = V(C^0) = V(1) = 2^n$. Thus, in the remaining of the paper, we will only consider the instances of the problem with $v_0 = 2^n$ and $v_1, \ldots, v_{2^\lambda-1} \in S$. For the other instances of the problem, it is obvious that no solution exists. Since it is more meaningful to consider a set of nonempty cubes $c_0, \ldots, c_{\lambda-1}$, we assume that for any $0 \le i \le \lambda - 1$, $v_{2^i} > 0$.

In our treatment, we find it convenient to represent a cube as a *cube-variable row vector* and a set of cubes as a *cube-variable matrix*. These are defined as follows.

Definition 3. Given a nonempty cube c on n variables $x_0, \ldots, x_{n-1}$, we represent it by a *cube-variable row vector* U of length n, whose elements are from the set $\{0, 1, *\}$. If the jth ($0 \le j \le n-1$) element $U_j = 1$, then the literal x_j appears in the cube c; if $U_j = 0$, then the literal $\bar{x}_j$ appears in the cube c; if $U_j = *$, then the cube c does not depend on the variable x_j, i.e., neither literal x_j nor literal $\bar{x}_j$ appears in the cube c.

Given a set of λ nonempty cubes $c_0, \ldots, c_{\lambda-1}$ on n variables $x_0, \ldots, x_{n-1}$, we represent them by a *cube-variable matrix* D of size $\lambda \times n$, so that the ith row of the matrix is the cube-variable row vector of c_i. □

For example, a set of two cubes $c_0 = x_0\bar{x}_1$ and $c_1 = \bar{x}_0 x_2$ is represented as a cube-variable matrix

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

Given a cube-variable row vector, the following simple lemma suggests how to obtain the number of minterms covered by the corresponding cube.

Lemma 1. *For a nonempty cube, if its cube-variable row vector contains k $*$'s, then the cube covers 2^k number of minterms.* □

Proof. Assume that the cube-variable row vector is $(a_1, \ldots, a_n)$ ($n \ge k$). Without loss of generality, we assume that the first $(n-k)$ entries of the row vector are not $*$'s and the last k entries of the row vector are $*$'s, i.e., $a_1, a_2, \ldots, a_{n-k} \in \{0, 1\}$ and $a_{n-k+1} = a_{n-k+2} = \cdots = a_n = *$. Then, the row vector covers 2^k minterms whose cube-variable row vectors are $(a_1, \ldots, a_{n-k}, 0, 0, \ldots, 0, 0), (a_1, \ldots, a_{n-k}, 0, 0, \ldots, 0, 1), \ldots, (a_1, \ldots, a_{n-k}, 1, 1, \ldots, 1, 1)$. □

In what follows, we will say that a cube-variable matrix satisfies the given intersection pattern if the corresponding set of cubes satisfies the intersection pattern.

We find that several operations on the cube-variable matrix will keep the intersection pattern unchanged. One operation relates to the negation operator defined below.

Definition 4. For a value a in $\{0, 1, *\}$, the negation of a is defined as follows:

$$\bar{a} = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{if } a = 1 \\ *, & \text{if } a = *. \end{cases}$$

The negation of a column vector (cube-variable matrix) is the element-wise negation of the column vector (matrix). □

An important property of a cube-variable matrix is that performing column permutation or column negation on the matrix keeps the intersection pattern unchanged, as stated by the following lemma.

Lemma 2. *Suppose that a cube-variable matrix D satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$. Then D' satisfies the same intersection pattern if D' is obtained from D by column permutation or column negation.* □

Proof. Assume that the intersection pattern of the cube-variable matrix D' is $(v'_0, \ldots, v'_{2^\lambda-1})$. We only need to show that for all $0 \le \Gamma \le 2^\lambda - 1$, $v_\Gamma = v'_\Gamma$.

Obviously, $v_0 = v'_0 = 2^n$, where n is the total number of variables. Now consider any $1 \le \Gamma \le 2^\lambda - 1$. Assume that $\Gamma = \sum_{i=0}^{r-1} 2^{l_i}$, where $1 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$. Denote the cube-variable matrix which consists of the row $l_0, l_1, \ldots, l_{r-1}$ of the matrix D as D_Γ and the cube-variable matrix which consists of the row $l_0, l_1, \ldots, l_{r-1}$ of the matrix D' as D'_Γ. Then, D'_Γ is obtained from D_Γ by column permutation or column negation. We consider two cases.

1. The case where there exists a column in D_Γ that contains both a 0 and a 1. In this case, there must exist a column in D'_Γ that contains both a 0 and a 1. Therefore, $v_\Gamma = v'_\Gamma = 0$.
2. The case where there exists no column in D_Γ that contains both a 0 and a 1. Assume that there are k columns in D_Γ of which all the entries are $*$'s. Then, we have $v_\Gamma = 2^k$. Since D'_Γ is obtained from D_Γ by column permutation or column negation, it has no column that contains both a 0 and a 1. Further, the number of columns in D'_Γ that have all the entries as $*$ is k. Thus, $v'_\Gamma = 2^k = v_\Gamma$.

Thus, we have proved that for all $0 \le \Gamma \le 2^\lambda - 1$, $v_\Gamma = v'_\Gamma$. □

Before we go through the details of our proposed solution, we will briefly talk about the basic idea of our solution. Our solution is a column-based method: synthesizing a cube-variable matrix is equivalent to determining what each column of the matrix should be. Since each entry of the matrix is in the set $\{0, 1, *\}$, each column, which has λ entries, has a total of 3^λ choices. However, we only need to consider a small subset of all 3^λ column choices as the candidate choices. One reason for this is because the negation of a column does not change the intersection pattern, as Lemma 2 indicates. Thus, for each pair of column choice and its negation, we only need to pick one as the candidate choice. Furthermore, by Lemma 2, since the order of the columns does not matter, we only need to determine the number of occurrences of each candidate column choice in the cube-variable matrix. We treat their numbers of occurrences as unknowns. We could establish a system of equations over those unknowns and the given intersection pattern. The λ-cube intersection problem can be solved by finding a non-negative solution to the system of equations.

3. A special case of the λ-cube intersection problem

In this section, we consider a special case in which $v_{2^\lambda-1} > 0$. We will study the necessary and sufficient condition on $(v_0, \ldots, v_{2^\lambda-1})$ so that there exists a set of λ cubes that satisfies the intersection pattern. For this purpose, we will assume that there exists a cube-variable matrix D to satisfy the given intersection pattern.

We argue that without loss of generality, we can assume that each entry of the cube-variable matrix is either 1 or ∗. Since $v_{2^\lambda-1} > 0$, we must have $\prod_{i=0}^{\lambda-1} c_i \neq 0$. Therefore, no column of the matrix D could simultaneously contain both a 0 and a 1; otherwise, $\prod_{i=0}^{\lambda-1} c_i = 0$. Consequently, each column of the matrix D contains either only 0's and ∗'s or only 1's and ∗'s. By Lemma 2, if we negate those columns of the matrix D that contain only 0's and ∗'s, then we obtain a new matrix D' which still satisfies the given intersection pattern. Note that the matrix D' only contains 1's and ∗'s. In this case, all the cubes are composed of uncomplemented literals x_i's and therefore, the union of these cubes is a positive unate Boolean function [4].

Since the matrix only contains 1's and ∗'s, each column of the matrix is a length-λ vector composed of either 1 or ∗. There are 2^λ different length-λ vectors that are composed of either 1 or ∗. We denote them as $\psi_0, \psi_1, \ldots, \psi_{2^\lambda-1}$.

Definition 5. Given any $0 \leq \Gamma \leq 2^\lambda - 1$, suppose that $\Gamma = \sum_{i=0}^{\lambda-1} \gamma_i 2^i$, where $\gamma_i \in \{0, 1\}$. Define ψ_Γ to be a column vector of length λ that is composed of either 1 or ∗, such that the ith element $(0 \leq i \leq \lambda - 1)$ of it is

$$(\psi_\Gamma)_i = \begin{cases} 1, & \text{if } \gamma_i = 0 \\ *, & \text{if } \gamma_i = 1. \end{cases}$$

Define the set $\Psi = \{\psi_0, \psi_1, \ldots, \psi_{2^\lambda-1}\}$. □

For example, if $\lambda = 3$, then $\psi_1 = (*, 1, 1)^T$ and $\psi_5 = (*, 1, *)^T$.[1]

As Lemma 2 states, if there exists a cube-variable matrix D that satisfies the intersection pattern, then a matrix obtained by permuting the columns of the matrix D also satisfies the intersection pattern. Therefore, the order on the columns in the matrix does not matter. What matters is the number of times that each column pattern ψ_Γ occurs in the matrix. We define that number as z_Γ.

Definition 6. For any $0 \leq \Gamma \leq 2^\lambda - 1$, define z_Γ to be the number of occurrences of column pattern ψ_Γ in the cube-variable matrix. □

In the special case that $v_{2^\lambda-1} > 0$, we have the following theorem on the values v_Γ's in the intersection pattern.

Theorem 1. *Suppose that there exists a cube-variable matrix that satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$ and $v_{2^\lambda-1} > 0$. Then for any $0 \leq \Gamma \leq 2^\lambda - 1$, we have $v_\Gamma > 0$.* □

Proof. Suppose that the set of cubes that satisfies the intersection pattern is $\{c_0, \ldots, c_{\lambda-1}\}$. Based on Definition 2, for any $0 \leq \Gamma \leq 2^\lambda - 1$, we have $C^{2^\lambda-1} \subseteq C^\Gamma$, Therefore,

$$0 < v_{2^\lambda-1} = V(C^{2^\lambda-1}) \leq V(C^\Gamma) = v_\Gamma. \quad \square$$

As we stated in Section 2, for any $0 \leq \Gamma \leq 2^\lambda - 1$, $v_\Gamma \in S = \{s | s = 0 \text{ or } s = 2^k, k = 0, 1, \ldots, n\}$. Now since $v_\Gamma > 0$, we have $v_\Gamma = 2^{k_\Gamma}$, where $k_\Gamma \in \{0, 1, \ldots, n\}$, for all $0 \leq \Gamma \leq 2^\lambda - 1$. In what follows, we will establish a relation between z_Γ's, which are the numbers of occurrences of patterns ψ_Γ's in the cube-variable matrix, and k_Γ's, which are obtained from the intersection pattern. In order to state that relation, we first define the following relation between two numbers A and B.

Definition 7. Given two integers A and B, let their binary representations be $A = \sum_{i=0}^{k-1} a_i 2^i$ and $B = \sum_{i=0}^{k-1} b_i 2^i$, where $a_i, b_i \in \{0, 1\}$. We write $A \sqsupseteq B$ if for all $0 \leq i \leq k - 1$, $a_i \geq b_i$; we write $A \sqsubseteq B$ if for all $0 \leq i \leq k - 1$, $a_i \leq b_i$. □

With the help of the above definition, we can state a major result in this section.

Theorem 2. *If there exists a cube-variable matrix D that satisfies the intersection pattern, then for all $0 \leq L \leq 2^\lambda - 1$, we have*

$$k_L = \sum_{0 \leq \Gamma \leq 2^\lambda-1 : \Gamma \sqsupseteq L} z_\Gamma. \quad \square \tag{2}$$

[1] The superscript T here means the transpose of a vector.

Proof. Since the total number of columns in matrix D is n, we have $\sum_{\Gamma=0}^{2^\lambda-1} z_\Gamma = n$. Further, since $v_0 = 2^n$ (as we stated in Section 2), we have $k_0 = n$. Therefore,

$$\sum_{0\le\Gamma\le 2^\lambda-1:\Gamma\sqsupseteq 0} z_\Gamma = k_0.$$

Thus, Eq. (2) holds for $L = 0$.

Now consider any $1 \le L \le 2^\lambda - 1$. L can be represented as $L = \sum_{j=0}^{r-1} 2^{l_j}$, where $1 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$. Then, C^L represents a cube that is the intersection of the set of cubes $c_{l_0}, \ldots, c_{l_{r-1}}$, i.e., $C^L = \prod_{j=0}^{r-1} c_{l_j}$. Due to this relation, the ith entry in the cube-variable row vector of C^L is $*$ if and only if the ith column of the cube-variable matrix D has $*$'s on the rows $l_0, l_1, \ldots, l_{r-1}$. Therefore, the number of $*$'s in the cube-variable row vector of C^L is the number of columns in D whose entries on the rows $l_0, l_1, \ldots, l_{r-1}$ are all $*$'s, or, the sum of the numbers of occurrences of patterns ψ_Γ's in D with the l_0th, l_1th, …, l_{r-1}th entries all being $*$, i.e.,

$$\sum_{\substack{0\le\Gamma\le 2^\lambda-1:\\ (\psi_\Gamma)_{l_0}=\cdots=(\psi_\Gamma)_{l_{r-1}}=*}} z_\Gamma.$$

On the other hand, by Lemma 1, since $V(C^L) = v_L = 2^{k_L}$, it indicates that the number of $*$'s in the cube-variable row vector of C^L is k_L. Therefore, together with Definition 5, we have

$$k_L = \sum_{\substack{0\le\Gamma\le 2^\lambda-1:\\ (\psi_\Gamma)_{l_0}=\cdots=(\psi_\Gamma)_{l_{r-1}}=*}} z_\Gamma = \sum_{\substack{0\le\Gamma\le 2^\lambda-1,\\ \Gamma=\sum_{i=0}^{\lambda-1}\gamma_i 2^i:\\ \gamma_{l_0}=\cdots=\gamma_{l_{r-1}}=1}} z_\Gamma. \tag{3}$$

By Definition 7, we can rewrite Eq. (3) as

$$k_L = \sum_{0\le\Gamma\le 2^{\lambda-1}:\Gamma\sqsupseteq L} z_\Gamma. \quad \square$$

Example 3. Consider the following 4 cubes on 6 variables $x_0, x_1, \ldots, x_5$:

$$c_0 = x_0x_1x_2x_4, \qquad c_1 = x_3, \qquad c_2 = x_0, \qquad c_3 = x_0x_1x_3x_4x_5.$$

Their cube-variable matrix is

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

Based on Definition 5, the above matrix can be represented in terms of ψ_Γ as

$$\begin{bmatrix} \psi_2 & \psi_6 & \psi_{14} & \psi_5 & \psi_6 & \psi_7 \end{bmatrix}.$$

Thus, we can get the number of occurrences of each pattern ψ_Γ in the matrix as

$$z_2 = 1, \quad z_5 = 1, \quad z_6 = 2, \quad z_7 = 1, \quad z_{14} = 1,$$
$$z_\Gamma = 0, \quad \text{for } \Gamma = 0, 1, 3, 4, 8, 9, 10, 11, 12, 13, 15.$$

It can be shown that Eq. (2) holds for all $0 \le L \le 15$. As an example, now we verify that Eq. (2) holds for $L = 6$. First, notice that $C^6 = c_1c_2 = x_0x_3$. Therefore, $v_6 = V(C^6) = 2^4 = 16$ and $k_6 = 4$. On the other hand,

$$\sum_{0\le\Gamma\le 15:\Gamma\sqsupseteq 6} z_\Gamma = z_6 + z_7 + z_{14} + z_{15} = 4,$$

which indicates that

$$\sum_{0\le\Gamma\le 15:\Gamma\sqsupseteq 6} z_\Gamma = k_6.$$

Therefore, Eq. (2) holds for $L = 6$. $\square$

Note that Eq. (2) is a linear equation on $z_0, \ldots, z_{2^\lambda-1}$ and holds for all $0 \le L \le 2^\lambda - 1$. Therefore, we can derive a system of 2^λ linear equations on unknowns $z_0, \ldots, z_{2^\lambda-1}$:

$$\sum_{0\le\Gamma\le 2^\lambda-1:\Gamma\sqsupseteq L} z_\Gamma = k_L, \quad \text{for } L = 0, 1, \ldots, 2^\lambda - 1. \tag{4}$$

We can represent the above system of linear equations in matrix form, as shown by the following theorem.

Theorem 3. *Let vector* $\vec{k} = (k_0, \ldots, k_{2^\lambda-1})^T$ *and vector* $\vec{z} = (z_0, \ldots, z_{2^\lambda-1})^T$. *Then we can represent the system of* 2^λ *linear equations shown in Eq. (4) in matrix form as*

$$R_\lambda \vec{z} = \vec{k}, \tag{5}$$

where R_λ *is a* $2^\lambda \times 2^\lambda$ *square matrix defined recursively as follows:*

$$R_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad R_i = \begin{bmatrix} R_{i-1} & R_{i-1} \\ 0 & R_{i-1} \end{bmatrix}, \quad \text{for } i = 2, \ldots, \lambda. \quad \square$$

Proof. For convenience, we use $\vec{z}[j, k] (0 \le j \le k \le 2^\lambda - 1)$ to represent the column vector $(z_j, \ldots, z_k)^T$.

We claim that given any $1 \le i \le \lambda$, the set of 2^i linear expressions

$$\sum_{0\le\Gamma\le 2^i-1:\Gamma\sqsupseteq L} z_\Gamma, \quad \text{for } L = 0, 1, \ldots, 2^i - 1$$

can be represented in matrix form as

$$R_i \vec{z}[0, 2^i - 1].$$

We prove this claim by induction on i.

Base case: When $i = 1$, the set of 2 linear expressions

$$\begin{cases} \sum_{0\le\Gamma\le 1:\Gamma\sqsupseteq 0} z_\Gamma \\ \sum_{0\le\Gamma\le 1:\Gamma\sqsupseteq 1} z_\Gamma \end{cases}$$

is

$$\begin{cases} z_0 + z_1 \\ z_1 \end{cases}.$$

Therefore, in the matrix form, the set of expressions can be represented as $R_1 \vec{z}[0, 1]$.

Inductive step: Assume that the claim holds for i. Now consider the set of 2^{i+1} linear expressions

$$\sum_{0\le\Gamma\le 2^{i+1}-1:\Gamma\sqsupseteq L} z_\Gamma, \quad \text{for } L = 0, 1, \ldots, 2^{i+1} - 1. \tag{6}$$

For any $0 \le L \le 2^{i+1} - 1$, we have

$$\sum_{\substack{0\le\Gamma\le 2^{i+1}-1: \\ \Gamma\sqsupseteq L}} z_\Gamma = \sum_{\substack{0\le\Gamma\le 2^i-1: \\ \Gamma\sqsupseteq L}} z_\Gamma + \sum_{\substack{2^i\le\Gamma\le 2^{i+1}-1: \\ \Gamma\sqsupseteq L}} z_\Gamma = \sum_{\substack{0\le\Gamma\le 2^i-1: \\ \Gamma\sqsupseteq L}} z_\Gamma + \sum_{\substack{0\le\Gamma\le 2^i-1: \\ (\Gamma+2^i)\sqsupseteq L}} z_{(\Gamma+2^i)}. \tag{7}$$

Now consider the first 2^i expressions of (6). In this case, $0 \le L \le 2^i - 1$. It is not hard to see that

$$\{\Gamma | 0 \le \Gamma \le 2^i - 1, (\Gamma + 2^i) \sqsupseteq L\} = \{\Gamma | 0 \le \Gamma \le 2^i - 1, \Gamma \sqsupseteq L\}.$$

Thus, Eq. (7) can be rewritten as

$$\sum_{0\le\Gamma\le 2^{i+1}-1:\Gamma\sqsupseteq L} z_\Gamma = \sum_{0\le\Gamma\le 2^i-1:\Gamma\sqsupseteq L} z_\Gamma + \sum_{0\le\Gamma\le 2^i-1:\Gamma\sqsupseteq L} z_{(\Gamma+2^i)}.$$

By the induction hypothesis, the first 2^i expressions of (6)

$$\sum_{0\le\Gamma\le 2^{i+1}-1:\Gamma\sqsupseteq L} z_\Gamma, \quad \text{for } L = 0, \ldots, 2^i - 1$$

can be represented in matrix form as

$$R_i\vec{z}[0, 2^i - 1] + R_i\vec{z}[2^i, 2^{i+1} - 1]. \tag{8}$$

Now consider the last 2^i expressions of (6). In this case, $2^i \le L \le 2^{i+1} - 1$. It is not hard to see that

$$\{\Gamma | 0 \le \Gamma \le 2^i - 1, \Gamma \sqsupseteq L\} = \phi,$$
$$\{\Gamma | 0 \le \Gamma \le 2^i - 1, (\Gamma + 2^i) \sqsupseteq L\} = \{\Gamma | 0 \le \Gamma \le 2^i - 1, \Gamma \sqsupseteq (L - 2^i)\}.$$

Therefore, Eq. (7) can be rewritten as

$$\sum_{0\le\Gamma\le 2^{i+1}-1:\Gamma\sqsupseteq L} z_\Gamma = \sum_{0\le\Gamma\le 2^{i}-1:\Gamma\sqsupseteq (L-2^i)} z_{(\Gamma+2^i)}.$$

Note that since $2^i \le L \le 2^{i+1} - 1$, we have $0 \le L - 2^i \le 2^i - 1$. By the induction hypothesis, the last 2^i expressions of (6)

$$\sum_{0\le\Gamma\le 2^{i+1}-1:\Gamma\sqsupseteq L} z_\Gamma, \quad \text{for } L = 2^i, \ldots, 2^{i+1} - 1$$

can be represented in matrix form as

$$R_i\vec{z}[2^i, 2^{i+1} - 1]. \tag{9}$$

Based on Eqs. (8) and (9), the set of linear expressions

$$\sum_{0\le\Gamma\le 2^{i+1}-1:\Gamma\sqsupseteq L} z_\Gamma, \quad \text{for } L = 0, \ldots, 2^{i+1} - 1$$

can be represented in matrix form as

$$\begin{bmatrix} R_i & R_i \\ 0 & R_i \end{bmatrix} \begin{bmatrix} \vec{z}[0, 2^i - 1] \\ \vec{z}[2^i, 2^{i+1} - 1] \end{bmatrix} = R_{i+1}\vec{z}[0, 2^{i+1} - 1].$$

Therefore, the claim holds for $i + 1$. Thus, by induction, the claim holds for all $i = 1, 2, \ldots, \lambda$.

Thus, the system of linear equations

$$\sum_{0\le\Gamma\le 2^{\lambda}-1:\Gamma\sqsupseteq L} z_\Gamma = k_L, \quad \text{for } L = 0, 1, \ldots, 2^\lambda - 1$$

can be represented in matrix form as $R_\lambda\vec{z} = \vec{k}$. □

It is not hard to see that $\det(R_\lambda) = 1$. Therefore, R_λ is invertible. Based on Theorem 3, we can obtain the numbers of occurrences of all patterns ψ_Γ's in the matrix D as

$$\vec{z} = R_\lambda^{-1}\vec{k}. \tag{10}$$

The following lemma shows what the form of R_λ^{-1} is.

Lemma 3. *$R_1^{-1}, \ldots, R_\lambda^{-1}$, the inverses of the matrices $R_1, \ldots, R_\lambda$ defined in Theorem 3, have a recursive structure shown below:*

$$R_1^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \qquad R_i^{-1} = \begin{bmatrix} R_{i-1}^{-1} & -R_{i-1}^{-1} \\ 0 & R_{i-1}^{-1} \end{bmatrix} \quad \textit{for } i = 2, \ldots, \lambda. \quad \square$$

Proof. We only need to show that for $i = 1, \ldots, \lambda$, $R_i^{-1}R_i = I_{2^i}$, where I_{2^i} is a $2^i \times 2^i$ identity matrix. We prove this claim by induction on i.

Base case: When $i = 1$,

$$R_1^{-1}R_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Inductive step: Assume the claim holds for i. Then, based on the induction hypothesis,

$$R_{i+1}^{-1}R_{i+1} = \begin{bmatrix} R_i^{-1} & -R_i^{-1} \\ 0 & R_i^{-1} \end{bmatrix} \begin{bmatrix} R_i & R_i \\ 0 & R_i \end{bmatrix} = \begin{bmatrix} I_{2^i} & 0 \\ 0 & I_{2^i} \end{bmatrix} = I_{2^{i+1}}.$$

Therefore, the claim holds for $i + 1$. Thus, by induction, the claim holds for all $i = 1, \ldots, \lambda$. □

Therefore, given an intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$, we can get $z_0, \ldots, z_{2^\lambda-1}$ as $(z_0, \ldots, z_{2^\lambda-1})^T = R_\lambda^{-1}(k_0, \ldots, k_{2^\lambda-1})^T$, where $k_i = \log_2 v_i$.

Since for any $0 \le \Gamma \le 2^\lambda - 1$, z_Γ is the number of occurrences of ψ_Γ in the matrix D, it must be a non-negative integer. By Lemma 3, R_λ^{-1} is an integer matrix. Therefore, $z_0, \ldots, z_{2^\lambda-1}$ are always integers. Thus, a necessary condition for the existence of a cube-variable matrix to satisfy the given intersection pattern is that the vector $\vec{z} = R_\lambda^{-1}\vec{k}$ has all entries non-negative. On the other hand, from Eq. (5), we can see that the intersection pattern $(v_0, \ldots, v_{2^\lambda-1}) = (2^{k_0}, \ldots, 2^{k_{2^\lambda-1}})$ only depends on $z_0, \ldots, z_{2^\lambda-1}$. Therefore, as long as the vector $\vec{z} = R_\lambda^{-1}\vec{k}$ has all entries non-negative, there exists a cube-variable matrix that satisfies the given intersection pattern. Such a matrix contains z_Γ columns of column pattern ψ_Γ, for each $\Gamma = 0, \ldots, 2^\lambda - 1$. In summary, we have the following corollary.

Corollary 1. *The necessary and sufficient condition for the existence of a cube-variable matrix to satisfy a given intersection pattern* $(v_0, \ldots, v_{2^\lambda-1})$ *is that the vector* $\vec{z} = R_\lambda^{-1}\vec{k}$ *has all entries non-negative, where* $\vec{k} = (k_0, \ldots, k_{2^\lambda-1})^T = (\log_2(v_0), \ldots, \log_2(v_{2^\lambda-1}))^T$ *and* R_λ^{-1} *is defined in Lemma 3.* □

Example 4. Given $v_0 = 32$, $v_1 = 16$, $v_2 = 16$, $v_3 = 8$, $v_4 = 8$, $v_5 = 4$, $v_6 = 4$, and $v_7 = 2$, determine whether there exists a set of three cubes c_0, c_1, and c_2 on 5 variables that satisfies the intersection pattern $(v_0, \ldots, v_7)$.

Solution: From the given conditions, we have

$$\vec{k} = (5, 4, 4, 3, 3, 2, 2, 1)^T.$$

Since

$$R_3^{-1} = \begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

then by Eq. (10), we get

$$\vec{z} = (0, 0, 0, 2, 0, 1, 1, 1)^T.$$

Therefore, there are two ψ_3's, one ψ_5, one ψ_6, and one ψ_7 in the cube-variable matrix of the set of cubes c_0, c_1, and c_2. One realization of the cube-variable matrix is

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

and the corresponding cubes are $c_0 = x_3$, $c_1 = x_2$, and $c_2 = x_0x_1$. It can be verified that these three cubes satisfy the given intersection pattern. □

4. General λ-cube intersection problem

In this section, we consider the more general situation where $v_{2^\lambda-1} \ge 0$. In what follows, we will assume that there exists a cube-variable matrix D that satisfies the given intersection pattern. We first give an overview of our solution to the general case problem.

4.1. Overview of our solution

In the general case, the cube-variable matrix consists of 0, 1, and $*$; so does each column of the matrix. There are a total of 3^λ different choices of patterns for each column. However, not all combinations of 0, 1 and $*$ as a column vector can appear in the matrix. For example, if the given intersection pattern indicates that $c_i \cdot c_j \neq 0$, then those column patterns that have a 0 at the ith entry and a 1 at the jth entry cannot be present in the matrix. On the other hand, some kinds of column patterns must be present at least once in the matrix. For example, if the given intersection pattern indicates that $c_i \cdot c_j = 0$, then at least one of the column patterns that have a 0 at the ith entry and a 1 at the jth entry or have a 1 at the ith entry and a 0 at the jth entry must be present in the matrix.

In Section 4.2, we show what kinds of column patterns can be presented in the cube-variable matrix. We introduce the *representative compatible column pattern set* in Definition 12. We further define in Definition 13 a set F as the union of the representative compatible column pattern sets and the set Ψ (see Definition 5). Then, we present Lemma 5, which states that only those column patterns in the set F are needed to construct the cube-variable matrix.

In Section 4.3, we present Theorems 4 and 5 which give two necessary conditions on the numbers $v_\Gamma > 0$ in the given intersection pattern for the existence of a cube-variable matrix to satisfy the given intersection pattern.

In Section 4.4, we present our solution to the general λ-cube intersection problem. The idea is same as that used in solving the special case problem: we establish the numerical relations between the given intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$ and the numbers of times that the column patterns in the set F appear in the cube-variable matrix; the λ-cube intersection problem is then solved based on these relations. For this purpose, we first link the general case problem to the special case problem by defining the *root cube-variable matrix* in Definition 15. The root cube-variable matrix contain only 1's and $*$'s. Thus, we could define the numbers z_Γ (see Definition 6) on the root cube-variable matrix. We show in Theorem 6 a system of linear equations between the numbers z_Γ and the given intersection pattern. Then, we show in Theorem 7 a set of linear inequalities on the numbers z_Γ and the numbers of occurrences of the representative column patterns in the cube-variable matrix. Finally, we show the main result of this paper in Theorem 8, which states that the combination of Theorems 4–7 gives a necessary and sufficient condition for the existence of a cube-variable matrix to satisfy the given intersection pattern. The proof of Theorem 8 also indicates a way to synthesize a cube-variable matrix to satisfy the given intersection pattern.

4.2. The set of column patterns to compose the cube-variable matrix

In this section, we will show what kinds of column patterns can be presented in the matrix. Then, we will argue that we only need to focus on a subset of the total 3^λ column patterns to construct a cube-variable matrix.

First, we give a few definitions. In the general situation, some of the values v_Γ's in the intersection pattern are zero and the others are positive. Based on their values, we can split their indices into the following two sets.

Definition 8. Let the set P be the set of numbers Γ such that $v_\Gamma > 0$ and let the set Z be the set of numbers Γ such that $v_\Gamma = 0$, i.e.,

$$P = \{\Gamma | 0 \le \Gamma \le 2^\lambda - 1 \text{ and } v_\Gamma > 0\},$$
$$Z = \{\Gamma | 0 \le \Gamma \le 2^\lambda - 1 \text{ and } v_\Gamma = 0\}. \quad \square$$

From the definition of P and Z, we have the following lemma, which gives a necessary condition on the existence of λ cubes to satisfy the given intersection pattern.

Lemma 4. *Suppose that a set of λ cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the given intersection pattern. Then, for any $\Gamma \in P$, $C^\Gamma \neq 0$ and for any $\Gamma \in Z$, $C^\Gamma = 0$.* $\square$

Proof. Since the set of λ cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the given intersection pattern, we have that for any $0 \le \Gamma \le 2^\lambda - 1$, $V(C^\Gamma) = v_\Gamma$. By Definition 8, for any $\Gamma \in P$, $V(C^\Gamma) = v_\Gamma > 0$, which indicates that $C^\Gamma \neq 0$; for any $\Gamma \in Z$, $V(C^\Gamma) = v_\Gamma = 0$, which indicates that $C^\Gamma = 0$. $\square$

For any $\Gamma \in P$, since $v_\Gamma > 0$, we define a number k_Γ as follows:

Definition 9. For any $\Gamma \in P$, define $k_\Gamma = \log_2(v_\Gamma)$. $\square$

As we stated in Section 2, for any $0 \le \Gamma \le 2^\lambda - 1$, $v_\Gamma \in S = \{s | s = 0 \text{ or } s = 2^k, k = 0, 1, \ldots, n\}$. Therefore, for any $\Gamma \in P$, k_Γ is an integer and $k_\Gamma \in \{0, 1, \ldots, n\}$. Note that since $v_0 = 2^n$, we have $k_0 = n$.

We further define a number of subsets of the sets P and Z based on the number of ones in the binary representation of a number Γ.

Definition 10. For an integer $a \ge 0$, define $\|a\|$ to be the number of ones in the binary representation of a. More formally, suppose that a can be represented as $a = \sum_{i=0}^{k-1} a_i 2^i$ with all $a_i \in \{0, 1\}$. Then, $\|a\| = \sum_{i=0}^{k-1} a_i$.

For any $0 \le i \le \lambda$, let the set P_i be the set of numbers Γ such that the number of ones in the binary representation of Γ is i and $v_\Gamma > 0$; let the set Z_i be the set of Γ such that the number of ones in the binary representation of Γ is i and $v_\Gamma = 0$, i.e.,

$$P_i = \{\Gamma | 0 \le \Gamma \le 2^\lambda - 1, \|\Gamma\| = i, \text{ and } v_\Gamma > 0\},$$
$$Z_i = \{\Gamma | 0 \le \Gamma \le 2^\lambda - 1, \|\Gamma\| = i, \text{ and } v_\Gamma = 0\}. \quad \square$$

In our treatment, two important sets are P_2 and Z_2. The set P_2 indicates all pairs of cubes which are not disjoint and the set Z_2 indicates all pairs of cubes which are disjoint.

Example 5. Consider a 4-cube intersection problem with $v_0 = 64, v_1 = 4, v_2 = 8, v_3 = 2, v_4 = 16, v_5 = 2, v_6 = 0, v_7 = 0, v_8 = 8, v_9 = 1, v_{10} = 0, v_{11} = 0, v_{12} = 0, v_{13} = 0, v_{14} = 0, v_{15} = 0$. In binary representation, those indices $0 \leq \Gamma \leq 15$ with $\|\Gamma\| = 2$ are $(0011)_2$, $(0101)_2$, $(0110)_2$, $(1001)_2$, $(1010)_2$, $(1100)_2$. From the values of v_Γ's, we can obtain $P_2 = \{(0011)_2, (0101)_2, (1001)_2\}$ and $Z_2 = \{(0110)_2, (1010)_2, (1100)_2\}$. This indicates that the pairs of cubes (c_0, c_1), (c_0, c_2), and (c_0, c_3) are non-disjoint; the pairs of cubes (c_1, c_2), (c_1, c_3), and (c_2, c_3) are disjoint. □

Now, we are ready to show what kinds of column patterns can be present in the cube-variable matrix. It depends on which pairs of cubes should be disjoint and which pairs of cubes should be non-disjoint. In other words, it depends on the sets P_2 and Z_2. The intuition is that if the given intersection pattern indicates that $c_i \cdot c_j \neq 0$, then those column patterns that have a 0 at the ith entry and a 1 at the jth entry cannot be present in the matrix. On the other hand, some kinds of column patterns must be present at least once in the matrix. For example, if the given intersection pattern indicates that $c_i \cdot c_j = 0$, then at least one of the column patterns that have a 0 at the ith entry and a 1 at the jth entry or have a 1 at the ith entry and a 0 at the jth entry must be present in the matrix.

In what follows, we will first introduce the *compatible column pattern set* for a number $\Gamma \in Z_2$. Based on that, we will further introduce the *representative compatible column pattern set* for a number $\Gamma \in Z_2$. Later on, we will show that the column patterns in the representative compatible column pattern set for each $\Gamma \in Z_2$ can be present in the cube-variable matrix.

Definition 11. Suppose that $\Gamma \in Z_2$ and $\Gamma = 2^i + 2^j$, where $0 \leq i < j \leq \lambda - 1$. The compatible column pattern set for Γ is the set of column vectors W of length λ with entries being either 0, 1, or $*$, such that

1. $(W_i, W_j) = (0, 1)$ or $(1, 0)$,
2. for any number $L \in P_2$ such that $L = 2^k + 2^l$, where $0 \leq k < l \leq \lambda - 1$, the situation that $(W_k, W_l) = (0, 1)$ or $(1, 0)$ does not happen. □

It is not hard to see that if a cube-variable column vector is in the compatible column pattern set for a $\Gamma \in Z_2$, then the negation of that cube-variable column vector is also in that set. Therefore, we define the *representative compatible column pattern set* as follows.

Definition 12. The representative compatible column pattern set ρ_Γ for $\Gamma \in Z_2$ is a set of cube-variable column vectors in the compatible column pattern set for Γ with their first non-$*$ entry being 0. □

Example 6. Consider the previous 4-cube intersection problem shown in Example 5. We have derived that

$$P_2 = \{(0011)_2, (0101)_2, (1001)_2\},$$
$$Z_2 = \{(0110)_2, (1010)_2, (1100)_2\}.$$

Now we will derive the compatible column pattern set for $\Gamma = (0110)_2 \in Z_2$. Note that in our representation, when we represent an intersection of the cubes using the notation C^Γ, where the binary representation of Γ is $(\gamma_{\lambda-1} \ldots \gamma_0)_2$, the **rightmost** bit in the binary representation of Γ corresponds to the first cube c_0, while the **leftmost** bit corresponds to the last cube $c_{\lambda-1}$. However, for a cube-variable column vector $W = (W_0, W_1, \ldots, W_{\lambda-1})^T$, the **leftmost** entry in this transposed representation corresponds to the first cube c_0, while the **rightmost** entry corresponds to the last cube $c_{\lambda-1}$.

Based on Definition 11, a vector $W = (W_0, W_1, W_2, W_3)^T$ in the compatible column pattern set for $\Gamma = (0110)_2 \in Z_2$ should satisfy that

1. $(W_1, W_2) = (0, 1)$ or $(1, 0)$,
2. the following six situations, which are obtained based on the set P_2, do not happen: $(W_0, W_1) = (0, 1)$, $(W_0, W_1) = (1, 0)$, $(W_0, W_2) = (0, 1)$, $(W_0, W_2) = (1, 0)$, $(W_0, W_3) = (0, 1)$, and $(W_0, W_3) = (1, 0)$.

If $(W_1, W_2) = (0, 1)$, then W_0 can be neither 0 nor 1; otherwise, it violates the second condition above. Thus, W_0 can only be $*$. Similarly, W_0 can only be $*$ if $(W_1, W_2) = (1, 0)$. In both cases, W_3 can be 0, 1, or $*$.

Thus, the compatible column pattern set for $\Gamma = (0110)_2 \in Z_2$ is

$$\{(*, 0, 1, 0)^T, (*, 0, 1, 1)^T, (*, 0, 1, *)^T, (*, 1, 0, 0)^T, (*, 1, 0, 1)^T, (*, 1, 0, *)^T\}.$$

Based on Definition 12, the representative compatible column pattern set for $\Gamma = (0110)_2$ is

$$\{(*, 0, 1, 0)^T, (*, 0, 1, 1)^T, (*, 0, 1, *)^T\}. \quad \square$$

Definition 13. We define the set Y as the union of the representative compatible column pattern sets ρ_Γ for all $\Gamma \in Z_2$, i.e., $Y = \bigcup_{\Gamma \in Z_2} \rho_\Gamma$. We define the set $F = Y \cup \Psi$, where Ψ is given in Definition 5. □

Now we are going to state an important claim in this section, which says that we only need to focus on those column patterns in the set F to construct a cube-variable matrix that satisfies the intersection pattern.

Lemma 5. *If there exists a cube-variable matrix D that satisfies the given intersection pattern, then there exists another matrix D' which also satisfies the given intersection pattern and each column of which is in the set F.* □

Proof. First, we argue that for any column of D which contains both a 0 and a 1, the column is in the compatible column pattern set for a certain $\Gamma \in Z_2$.

Suppose that a column $r (0 \le r \le n-1)$ of D has the ith entry being 0 and the jth entry being 1, where $0 \le i, j \le \lambda - 1$ and $i \ne j$. Then, $c_i \cdot c_j = 0$. Since the matrix D satisfies the given intersection pattern, we have $v_{2^i+2^j} = V(c_i \cdot c_j) = 0$. Therefore, the number $2^i + 2^j$ is in the set Z_2. Now consider any $L \in P_2$. Suppose that $L = 2^k + 2^l$, where $0 \le k < l \le \lambda - 1$. Since a necessary condition for the cube-variable matrix to satisfy the given intersection pattern is that for $L \in P_2$, $C^L \ne 0$, thus the situation that $D_{kr} = 0$ and $D_{lr} = 1$ or $D_{kr} = 1$ and $D_{lr} = 0$ cannot happen. Therefore, the column r of D is in the compatible column pattern set for the number $(2^i + 2^j) \in Z_2$.

We can construct a D' from D as follows. For any column $0 \le r \le \lambda - 1$:

1. If $D_{\cdot r}$ contains only 1's and $*$'s, we let $D'_{\cdot r}$ be $D_{\cdot r}$. Then $D'_{\cdot r}$ is in the set Ψ.
2. If $D_{\cdot r}$ contains only 0's and $*$'s, we let $D'_{\cdot r}$ be the negation of the column $D_{\cdot r}$. Then $D'_{\cdot r}$ is in the set Ψ.
3. If $D_{\cdot r}$ contains both a 0 and a 1 and the first non-$*$ entry of $D_{\cdot r}$ is 0, we let $D'_{\cdot r}$ be $D_{\cdot r}$. Then, there exists a $\Gamma \in Z_2$ such that $D'_{\cdot r}$ is in the compatible column pattern set for Γ. Further, since the first non-$*$ entry of $D'_{\cdot r}$ is 0, $D'_{\cdot r}$ is in the representative compatible column pattern set for Γ, ρ_Γ.
4. If $D_{\cdot r}$ contains both a 0 and a 1 and the first non-$*$ entry of $D_{\cdot r}$ is 1, we let $D'_{\cdot r}$ be the negation of the column $D_{\cdot r}$. Then, there exists a $\Gamma \in Z_2$ such that $D'_{\cdot r}$ is in the compatible column pattern set for Γ. Further, since the first non-$*$ entry of $D'_{\cdot r}$ is 0, $D'_{\cdot r}$ is in the representative compatible column pattern set for Γ, ρ_Γ.

Then, by the above construction, each column of D' is in the set F. Further, D' is obtained from D by column negations. Thus, by Lemma 2, D' also satisfies the given intersection pattern. □

Based on Lemma 5, we only need to answer whether there exists a cube-variable matrix with columns from the set F to satisfy the given intersection pattern. The following lemma states that if such a matrix exists, then for each $\Gamma \in Z_2$, at least one of the column vectors from the set ρ_Γ must be present in that matrix.

Lemma 6. *If a cube-variable matrix D with columns from the set F satisfies the given intersection pattern, then for any $\Gamma \in Z_2$, there exists a column in D which is in the set ρ_Γ.* □

Proof. For any $\Gamma \in Z_2$, suppose that $\Gamma = 2^i + 2^j$, where $0 \le i < j \le \lambda - 1$. Since the cube-variable matrix satisfies the given intersection pattern, therefore, $V(c_i \cdot c_j) = v_\Gamma = 0$. Consequently, we have $c_i \cdot c_j = 0$. Thus, there must exist a column r in D, such that $D_{ir} = 0$ and $D_{jr} = 1$ or $D_{ir} = 1$ and $D_{jr} = 0$. Now consider any $L \in P_2$. Suppose that $L = 2^k + 2^l$, where $0 \le k < l \le \lambda - 1$. Since a necessary condition for the cube-variable matrix to satisfy the given intersection pattern is that for the $L \in P_2$, $C^L \ne 0$, the situation that $D_{kr} = 0$ and $D_{lr} = 1$ or $D_{kr} = 1$ and $D_{lr} = 0$ cannot happen. Therefore, the column r of D is in the compatible column pattern set for Γ. Further, since all the columns of D are in the set F, then column r must be in the representative compatible column pattern set for Γ, ρ_Γ. □

4.3. A few necessary conditions on the intersection pattern

In this section, we show a few necessary conditions on the given intersection pattern so that there exists a set of cubes to satisfy that intersection pattern. These statements will play an important role later in proving a necessary and sufficient condition for the existence of a set of cubes to satisfy the given intersection pattern. We first have the following theorem on those numbers $v_\Gamma > 0$ in the intersection pattern.

Theorem 4. *Suppose that there exists a set of λ cubes $c_0, \ldots, c_{\lambda-1}$ that satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$. For any $0 \le L \le 2^\lambda - 1$, if $v_L > 0$, then for any $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq L$, we have $v_\Gamma > 0$.* □

Proof. Based on Definitions 2 and 7, for any $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq L$, we have $C^L \subseteq C^\Gamma$. Therefore, $0 < v_L = V(C^L) \le V(C^\Gamma) = v_\Gamma$. □

For example, suppose that in a 4-cube intersection problem we are given $v_{11} > 0$. If there exist 4 cubes to satisfy the given intersection pattern, then since $v_{11} = V(c_0c_1c_3) > 0$, we must have $c_0c_1c_3 \ne 0$. Therefore, we have $v_1 = V(c_0) > 0$, $v_2 = V(c_1) > 0$, $v_8 = V(c_3) > 0$, $v_3 = V(c_0c_1) > 0$, $v_9 = V(c_0c_3) > 0$, and $v_{10} = V(c_1c_3) > 0$.

If a set of cubes is pairwise non-disjoint, then the intersection of all those cubes is also non-disjoint, as shown by the following lemma.

Lemma 7. *If a set of r cubes $c_{l_0}, \ldots, c_{l_{r-1}} (3 \le r \le \lambda, 0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1)$ is pairwise non-disjoint, i.e., for any $0 \le i < j \le r-1$, $c_{l_i} \cdot c_{l_j} \ne 0$, then their intersection $\prod_{i=0}^{r-1} c_{l_i}$ is nonempty.* □

Proof. By contraposition, suppose that $\prod_{i=0}^{r-1} c_{l_i} = 0$. Consider the cube-variable matrix on these r cubes. Since their intersection is empty, there exists a column in the matrix that contains both a 0 and a 1. The cube corresponding to the 0 entry and the cube corresponding to the 1 entry are disjoint. This contradicts the assumption that the given set of cubes is pairwise non-disjoint. □

Alternatively, Lemma 7 can be stated on the numbers v_Γ's. This gives another necessary condition for the existence of a set of cubes to satisfy the given intersection pattern.

Theorem 5. *Suppose that there exists a set of λ cubes $c_0, \ldots, c_{\lambda-1}$ that satisfies the given intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$. If a set of $r(3 \le r \le \lambda)$ numbers $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$ satisfies the condition that for any $0 \le i < j \le r-1$, $v_{(2^{l_i}+2^{l_j})} > 0$, then for the number $L = \sum_{i=0}^{r-1} 2^{l_i}$, $v_L > 0$.* □

Proof. Since the set of λ cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the given intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$, therefore for any $0 \le i < j \le r-1$, $V(c_{l_i} \cdot c_{l_j}) = v_{(2^{l_i}+2^{l_j})}$. Given that for any $0 \le i < j \le r-1$, $v_{(2^{l_i}+2^{l_j})} > 0$, we have $c_{l_i} \cdot c_{l_j} \ne 0$. This means that the set of r cubes $c_{l_0}, \ldots, c_{l_{r-1}} (3 \le r \le \lambda, 0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1)$ is pairwise non-disjoint. By Lemma 7, their intersection $\prod_{i=0}^{r-1} c_{l_i}$ is nonempty. Therefore, for the number $L = \sum_{i=0}^{r-1} 2^{l_i}$, $v_L = V\left(\prod_{i=0}^{r-1} c_{l_i}\right) > 0$. □

For example, suppose that in a 4-cube intersection problem we are given $v_3 > 0$, $v_9 > 0$, and $v_{10} > 0$. If there exist 4 cubes to satisfy the given intersection pattern, then since $V(c_0c_1) > 0$, $V(c_0c_3) > 0$, and $V(c_1c_3) > 0$, we must have $v_{11} = V(c_0c_1c_3) > 0$.

If both the conditions in Theorems 4 and 5 are satisfied, then we have the following lemma, which will play an important role in proving the necessary and sufficient condition later.

Lemma 8. *Suppose that an intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$ satisfies that*

1. *For any $0 \le L \le 2^\lambda - 1$, if $v_L > 0$, then for any $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq L$, $v_\Gamma > 0$.*
2. *For any set of $r(3 \le r \le \lambda)$ numbers $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$, if it satisfies the condition that for any $0 \le i < j \le r-1$, $v_{(2^{l_i}+2^{l_j})} > 0$, then for the number $L = \sum_{i=0}^{r-1} 2^{l_i}$, $v_L > 0$.*

From the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$, we obtain the sets P, Z, P_2, and Z_2 by applying Definitions 8 and 10.

Now suppose that a set of λ nonempty cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the condition that for any $\Gamma \in P_2$, $C^\Gamma \ne 0$ and for any $\Gamma \in Z_2$, $C^\Gamma = 0$. Then, this set of cubes will satisfy the condition that for any $\Gamma \in P$, $C^\Gamma \ne 0$ and for any $\Gamma \in Z$, $C^\Gamma = 0$. □

Proof. Based on Definitions 8 and 10, it is not hard to see that the sets $P_0, \ldots, P_\lambda$ form a partition of the set P and that the sets $Z_0, \ldots, Z_\lambda$ form a partition of the set Z. Thus, we only need to prove that for all $0 \le k \le \lambda$, the set of cubes satisfies the condition that for any $\Gamma \in P_k$, $C^\Gamma \ne 0$ and for any $\Gamma \in Z_k$, $C^\Gamma = 0$.

We first consider the case that $k = 0$. As we stated in Section 2, $v_0 = 2^n > 0$. Thus, $P_0 = \{0\}$ and $Z_0 = \phi$. Since $C^0 = 1$, thus we have that for any $\Gamma \in P_0$, $C^\Gamma \ne 0$. Since $Z_0 = \phi$, the statement that for any $\Gamma \in Z_0$, $C^\Gamma = 0$ also holds.

Now we consider the case that $k = 1$. Since we assumed in Section 2 that for any $0 \le i \le \lambda - 1$, $v_{2^i} > 0$, therefore, $P_1 = \{2^i | i = 0, \ldots, \lambda - 1\}$ and $Z_1 = \phi$. Since $c_0, \ldots, c_{\lambda-1}$ are all nonempty, thus we have that for any $\Gamma \in P_1$, $C^\Gamma \ne 0$. Since $Z_1 = \phi$, the statement that for any $\Gamma \in Z_1$, $C^\Gamma = 0$ also holds.

When $k = 2$, the statement that the set of cubes satisfies the condition that for any $\Gamma \in P_2$, $C^\Gamma \ne 0$ and for any $\Gamma \in Z_2$, $C^\Gamma = 0$ obviously holds.

Now we consider the case that $k \ge 3$. First, we consider any $L \in P_k$. Suppose that $L = \sum_{i=0}^{r-1} 2^{l_i}$, where $3 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$. Then, for any $0 \le i < j \le r-1$, $(2^{l_i} + 2^{l_j}) \sqsubseteq L$. Since $v_L > 0$ and $(2^{l_i} + 2^{l_j}) \sqsubseteq L$, based on the condition 1 on the intersection pattern, we have $v_{(2^{l_i}+2^{l_j})} > 0$. Since $\|2^{l_i} + 2^{l_j}\| = 2$, thus $(2^{l_i} + 2^{l_j}) \in P_2$. By the assumption that for any $\Gamma \in P_2$, $C^\Gamma \ne 0$, we have that $C^{(2^{l_i}+2^{l_j})} = c_{l_i} \cdot c_{l_j} \ne 0$. Note that the numbers i and j are arbitrary. Thus, the r cubes $c_{l_0}, \ldots, c_{l_{r-1}}$ are pairwise non-disjoint. By Lemma 7, then $C^L = \prod_{i=0}^{r-1} c_{l_i} \ne 0$. Therefore, for any $L \in P_k$, $C^L \ne 0$.

Now we consider any $L \in Z_k$. Suppose that $L = \sum_{i=0}^{r-1} 2^{l_i}$, where $3 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$. We argue that there exist two numbers $0 \le u < v \le r-1$, such that $v_{(2^{l_u}+2^{l_v})} = 0$. Otherwise, for any $0 \le i < j \le r-1$, $v_{(2^{l_i}+2^{l_j})} > 0$. Then, based on the condition 2 on the intersection pattern, we have $v_L > 0$. This contradicts the assumption that $L \in Z_k$. Thus, there exist two numbers $0 \le u < v \le r-1$, such that $v_{(2^{l_u}+2^{l_v})} = 0$. Since $\|2^{l_u} + 2^{l_v}\| = 2$, thus $(2^{l_u} + 2^{l_v}) \in Z_2$. By the assumption that for any $\Gamma \in Z_2$, $C^\Gamma = 0$, we have that $C^{(2^{l_u}+2^{l_v})} = c_{l_u} \cdot c_{l_v} = 0$. Thus, $C^L = \prod_{i=0}^{r-1} c_{l_i} = 0$. Therefore, for any $L \in Z_k$, $C^L = 0$. □

4.4. A necessary and sufficient condition

In this section, we will show a necessary and sufficient condition for the existence of a set of cubes to satisfy the given intersection pattern. As a byproduct, the proof provides a way of synthesizing a set of cubes to satisfy the given intersection pattern. Based on Lemma 5, we only need to consider cube-variable matrix that consists of column patterns in the set F (defined in Definition 13). The basic idea to solve the general case problem is similar to that applied in the special case — we will establish relations between the intersection pattern and the numbers of times that those column patterns of the set F occur in the cube-variable matrix.

First, we introduce an important concept: the *root column vector*.

Definition 14. Given a column vector W with each element in the set $\{0, 1, *\}$, define its *root column vector* $t(W)$ as the column vector obtained from W by replacing the 0 entries in W with 1's and keeping the other entries in W unchanged. □

For example, given a column vector $(0, 1, *, 0)^T$, its root column vector is $(1, 1, *, 1)^T$. The root column vector connects the column patterns in the set F to those in the set Ψ (defined in Definition 5). As we will show later, with the aid of the root column vector, we can establish a relation between those positive values in the intersection pattern (i.e., those v_Γ's for $\Gamma \in P$) and the numbers of times that those column patterns of the set F occur in the cube-variable matrix.

If we replace each column of a cube-variable matrix by its root column vector, we will obtain a *root cube-variable matrix* of the original matrix, defined below.

Definition 15. Given a cube-variable matrix D of a set of λ cubes $c_0, \ldots, c_{\lambda-1}$, we define its *root cube-variable matrix* $t(D)$ as the cube-variable matrix formed by replacing each column in D with its root column vector. The set of cubes $c'_0, \ldots, c'_{\lambda-1}$ corresponding to the root cube-variable matrix is called the set of *root cubes* to the original set of cubes. □

For example, the root cube-variable matrix of the matrix

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

The set of root cubes is $c'_0 = x_0x_1$ and $c'_1 = x_0x_2$.

Based on the definition of the set of root cubes, we have the following lemma.

Lemma 9. *Suppose that a set of cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$. Further, suppose that the root cubes to the cubes $c_0, \ldots, c_{\lambda-1}$ are $c'_0, \ldots, c'_{\lambda-1}$. Then, for any $\Gamma \in P$, we have $V(C'^{\Gamma}) = V(C^\Gamma) = v_\Gamma$.* □

Proof. If $\Gamma = 0$, then obviously, $V(C'^0) = V(C^0) = 2^n = v_0$. Now consider any $\Gamma \in P$ such that $\Gamma \neq 0$. By the definition of the set P, we have $v_\Gamma > 0$. Since the set of cubes $c_0, \ldots, c_{\lambda-1}$ satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$, we have $V(C^\Gamma) = v_\Gamma > 0$. Suppose that C^Γ represents the intersection of a set of cubes $c_{l_0}, \ldots, c_{l_{r-1}}$, where $1 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$. Then, the intersection of $c_{l_0}, \ldots, c_{l_{r-1}}$ is nonempty.

Let the cube-variable matrix corresponding to the set of cubes $c_{l_0}, \ldots, c_{l_{r-1}}$ be D_Γ and the cube-variable matrix corresponding to the set of cubes $c'_{l_0}, \ldots, c'_{l_{r-1}}$ be D'_Γ. Based on the definition of the root cubes, each column of the matrix D'_Γ contains only 1's and $*$'s. Therefore, the intersection of $c'_{l_0}, \ldots, c'_{l_{r-1}}$ is also nonempty.

It is not hard to see that D'_Γ is the root cube-variable matrix of D_Γ. Therefore, the two matrices D'_Γ and D_Γ have the same number of columns that contain all $*$'s, i.e., columns of the form $(*, *, \ldots, *)^T$. Now consider the number of $*$'s in the cube-variable row vector of the intersection C^Γ. Since C^Γ is nonempty, that number should be equal to the number of columns in the matrix D_Γ that contain all $*$'s. The same claim applies to C'^{Γ}. Therefore, the number of $*$'s in the cube-variable row vector of C'^{Γ} equals that in the cube-variable row vector of C^Γ. By Lemma 1, we have $V(C'^{\Gamma}) = V(C^\Gamma) = v_\Gamma$. □

Example 7. Consider the following 3 cubes on 3 variables x_0, x_1, x_2:

$$c_0 = x_0, \qquad c_1 = \bar{x}_0x_1, \qquad c_2 = x_1x_2.$$

They satisfy the intersection pattern

$$v_0 = 8, \quad v_1 = 4, \quad v_2 = 2, \quad v_3 = 0, \quad v_4 = 2, \quad v_5 = 1, \quad v_6 = 1, \quad v_7 = 0.$$

The set P defined on the above intersection pattern is

$$P = \{0, 1, 2, 4, 5, 6\}.$$

The root cubes correspond to c_0, c_1, c_2 are

$$c'_0 = x_0, \qquad c'_1 = x_0x_1, \qquad c'_2 = x_1x_2.$$

It is not hard to verify that for any $\Gamma \in P$, $V(C'^{\Gamma}) = V(C^\Gamma) = v_\Gamma$. For example, for $\Gamma = 6$, we have

$$V(C'^{\Gamma}) = V(c'_1c'_2) = 1 = V(C^\Gamma) = V(c_1c_2) = v_6. \quad \square$$

Since the root cube-variable matrix $t(D)$ only contains column patterns in the set Ψ (defined in Definition 5), we can apply the definition of z_Γ (shown in Definition 6) to $t(D)$, which is the number of occurrences of the column pattern ψ_Γ in the matrix $t(D)$. Further, for any $\Gamma \in P$, we can define k_Γ according to Definition 9. The following theorem characterizes the relation between z_Γ's and k_Γ's.

Theorem 6. *If there exists a cube-variable matrix D that satisfies a given intersection pattern* $(v_0, \ldots, v_{2^\lambda-1})$, *then for any* $L \in P$, *we have*

$$\sum_{0 \le \Gamma \le 2^\lambda - 1 : \Gamma \sqsupseteq L} z_\Gamma = k_L,$$

where z_Γ*'s are defined on the root cube-variable matrix* $t(D)$ *of D according to Definition 6 and* k_L*'s are defined in Definition 9.* □

Proof. Consider a set of cubes $c_0, \ldots, c_{\lambda-1}$ that satisfies the intersection pattern $(v_0, \ldots, v_{2^\lambda-1})$. By Lemma 9, the corresponding set of root cubes $c'_0, \ldots, c'_{\lambda-1}$ has the following property: for any $\Gamma \in P$, $V(C'^{\Gamma}) = v_\Gamma = 2^{k_\Gamma}$. By applying the same reasoning used in proving Theorem 2 to the cube-variable matrix $t(D)$ (which corresponds to the set of cubes $c'_0, \ldots, c'_{\lambda-1}$), we can prove the claim that for any $L \in P$,

$$\sum_{0 \le \Gamma \le 2^\lambda - 1 : \Gamma \sqsupseteq L} z_\Gamma = k_L. \quad \square$$

Example 8. Consider the set of 3 cubes on 3 variables x_0, x_1, x_2 shown in Example 7:

$$c_0 = x_0, \qquad c_1 = \bar{x}_0 x_1, \qquad c_2 = x_1 x_2.$$

Their cube-variable matrix D is

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

They satisfy the intersection pattern

$$v_0 = 8, \quad v_1 = 4, \quad v_2 = 2, \quad v_3 = 0, \quad v_4 = 2, \quad v_5 = 1, \quad v_6 = 1, \quad v_7 = 0.$$

The set P defined on the above intersection pattern is

$$P = \{0, 1, 2, 4, 5, 6\}.$$

The root cube-variable matrix $t(D)$ of the matrix D is

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

The matrix $t(D)$ can be represented in terms of ψ_Γ as

$$\begin{bmatrix} \psi_4 & \psi_1 & \psi_3 \end{bmatrix}.$$

We can get the number of occurrences of each pattern ψ_Γ in the matrix $t(D)$ as

$$z_1 = z_3 = z_4 = 1, \qquad z_0 = z_2 = z_5 = z_6 = z_7 = 0.$$

It is not hard to verify that for any $L \in P$,

$$\sum_{0 \le \Gamma \le 7 : \Gamma \sqsupseteq L} z_\Gamma = k_L.$$

For example, for $\Gamma = 1$, on the one hand, we have

$$\sum_{0 \le \Gamma \le 7 : \Gamma \sqsupseteq 1} z_\Gamma = z_1 + z_3 + z_5 + z_7 = 2.$$

On the other hand, from the intersection pattern, we have $k_1 = \log_2 v_1 = 2$. Therefore, we have

$$\sum_{0 \le \Gamma \le 7 : \Gamma \sqsupseteq 1} z_\Gamma = k_1. \quad \square$$

Based on the definition of the root column vector, we can regroup the elements in the set Y (defined in Definition 13) according to their root column vectors, which results in the following definition. The relation between the elements in the set Y and their root column vectors will be used later to derive a set of inequalities on the numbers of occurrences of the elements of the set F in the cube-variable matrix (See Theorem 7).

Definition 16. We define the set M to be the set of numbers $0 \le \Gamma \le 2^{\lambda} - 1$ such that there exists an element in the set Y, whose root column vector is ψ_{Γ}, i.e.,

$$M = \{\Gamma | 0 \le \Gamma \le 2^{\lambda} - 1, \text{ s.t. } \exists W \in Y \text{ s.t. } t(W) = \psi_{\Gamma}\}.$$

Define $\overline{M}$ as $\overline{M} = \{\Gamma | 0 \le \Gamma \le 2^{\lambda} - 1, \Gamma \notin M\}$.

For any $\Gamma \in M$, we define the set Y_{Γ} to be the set of elements in the set Y such that their root column vectors are ψ_{Γ}, i.e., $Y_{\Gamma} = \{W | W \in Y \text{ and } t(W) = \psi_{\Gamma}\}$. □

Notice that the sets $Y_{\Gamma} (\Gamma \in M)$ form a partition of the set Y.

Example 9. For the intersection pattern shown in Example 5, we have $P_2 = \{3, 5, 9\}$ and $Z_2 = \{6, 10, 12\}$. According to Definition 12, the representative compatible column pattern sets for the numbers in Z_2 are

$$\begin{aligned}
\rho_6 &= \{(*, 0, 1, 0)^T, (*, 0, 1, 1)^T, (*, 0, 1, *)^T\},\\
\rho_{10} &= \{(*, 0, 0, 1)^T, (*, 0, 1, 1)^T, (*, 0, *, 1)^T\},\\
\rho_{12} &= \{(*, 0, 1, 0)^T, (*, 0, 0, 1)^T, (*, *, 0, 1)^T\}.
\end{aligned}$$

Thus, according to Definition 13, we have

$$\begin{aligned}
Y &= \rho_6 \cup \rho_{10} \cup \rho_{12}\\
&= \{(*, 0, 1, 0)^T, (*, 0, 0, 1)^T, (*, 0, 1, 1)^T, (*, *, 0, 1)^T, (*, 0, *, 1)^T, (*, 0, 1, *)^T\}.
\end{aligned}$$

The set of root column vectors for all the vectors in Y is

$$\{(*, 1, 1, 1)^T, (*, *, 1, 1)^T, (*, 1, *, 1)^T, (*, 1, 1, *)^T\}.$$

Thus, based on the definition of the set M, we have

$$M = \{1, 3, 5, 9\}.$$

Based on the definition of the set Y_{Γ}, we have $Y_1 = \{(*, 0, 1, 0)^T, (*, 0, 0, 1)^T, (*, 0, 1, 1)^T\}$, $Y_3 = \{(*, *, 0, 1)^T\}$, $Y_5 = \{(*, 0, *, 1)^T\}$, and $Y_9 = \{(*, 0, 1, *)^T\}$. □

As we showed in Section 4.2, to solve the general case λ-cube intersection problem, we only need to consider cube-variable matrix that consists of column patterns in the set $F = Y \cup \Psi$. Indeed, we only need to determine the number of occurrences of each element of the set F in the cube-variable matrix. For this purpose, we define as a variable the number of occurrences of each element of the set Y in the cube-variable matrix. In fact, we define such a number on each partition Y_{Γ} of Y, as stated by the following definition.

Definition 17. For any $\Gamma \in M$, we let the $|Y_{\Gamma}|$ elements in the set Y_{Γ} be $\delta_{\Gamma,0}, \ldots, \delta_{\Gamma,|Y_{\Gamma}|-1}$. For any $\Gamma \in M$ and any $0 \le i \le |Y_{\Gamma}| - 1$, we define $w_{\Gamma,i}$ to be the number of occurrences of the column pattern $\delta_{\Gamma,i}$ in the cube-variable matrix. □

The following theorem establishes a set of linear inequalities on $w_{\Gamma,i}$'s and z_{Γ}'s, where the z_{Γ}'s are defined on the root cube-variable matrix according to Definition 6.

Theorem 7. *Suppose that there exists a cube-variable matrix D that satisfies the given intersection pattern, whose columns are from the set F. Then, we have that for any $\Gamma \in M$,*

$$\sum_{i=0}^{|Y_{\Gamma}|-1} w_{\Gamma,i} \le z_{\Gamma}, \tag{11}$$

where z_{Γ}'s are defined on the root cube-variable matrix $t(D)$ according to Definition 6. We also have that for any $L \in Z_2$,

$$\sum_{\substack{\Gamma \in M, 0 \le i \le |Y_{\Gamma}|-1:\\ \delta_{\Gamma,i} \in \rho_L}} w_{\Gamma,i} \ge 1, \tag{12}$$

i.e., at least one column in the matrix D belongs to the representative compatible column pattern set for L, ρ_L. □

Proof. Consider any $\Gamma \in M$. Based on the definition of $w_{\Gamma,i}$, $\sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i}$ is the total number of times that the column patterns in the set Y_Γ occur in the matrix D. For those columns belonging to the set Y_Γ, their root column vector is ψ_Γ. Therefore, in the root cube-variable matrix $t(D)$, the number of occurrences of the column pattern ψ_Γ must be larger than the total number of times that the column patterns in the set Y_Γ occur in the matrix D, i.e.,

$$z_\Gamma \geq \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i}.$$

By Lemma 6, for any $L \in Z_2$, there exists a column in D which is in the set ρ_L. Suppose that the column is of the form $\delta_{\Gamma^*,i^*} \in \rho_L$, where $\Gamma^* \in M$ and $0 \leq i^* \leq |Y_{\Gamma^*}| - 1$. Then, we have

$$1 \leq w_{\Gamma^*,i^*} \leq \sum_{\substack{\Gamma \in M, 0 \leq i \leq |Y_\Gamma|-1: \\ \delta_{\Gamma,i} \in \rho_L}} w_{\Gamma,i}. \quad \square$$

Example 10. For the intersection pattern given in Example 5, based on Definition 17 and the result shown in Example 9, we have

$$\delta_{1,0} = (*, 0, 1, 0)^T, \delta_{1,1} = (*, 0, 0, 1)^T, \delta_{1,2} = (*, 0, 1, 1)^T,$$
$$\delta_{3,0} = (*, *, 0, 1)^T, \delta_{5,0} = (*, 0, *, 1)^T, \delta_{9,0} = (*, 0, 1, *)^T.$$

The set of equations shown in Eq. (11) for all $\Gamma \in M$ in this example is

$$\begin{cases} w_{1,0} + w_{1,1} + w_{1,2} \leq z_1 \\ w_{3,0} \leq z_3 \\ w_{5,0} \leq z_5 \\ w_{9,0} \leq z_9. \end{cases}$$

Based on the representative compatible column pattern sets shown in Example 9, we have

$$\rho_6 = \{\delta_{1,0}, \delta_{1,2}, \delta_{9,0}\},$$
$$\rho_{10} = \{\delta_{1,1}, \delta_{1,2}, \delta_{5,0}\},$$
$$\rho_{12} = \{\delta_{1,0}, \delta_{1,1}, \delta_{3,0}\}.$$

Thus, the set of equations shown in Eq. (12) for all $L \in Z_2$ in this example is

$$\begin{cases} w_{1,0} + w_{1,2} + w_{9,0} \geq 1 \\ w_{1,1} + w_{1,2} + w_{5,0} \geq 1 \\ w_{1,0} + w_{1,1} + w_{3,0} \geq 1. \quad \square \end{cases}$$

Finally, combining the conditions of Theorems 4–7, we can derive a major result in this section, which gives a necessary and sufficient condition for the existence of a cube-variable matrix to satisfy the given intersection pattern.

Theorem 8. *There exists a cube-variable matrix D that satisfies the given intersection pattern* $(v_0, \ldots, v_{2^\lambda-1})$ *if and only if*

1. *For any* $0 \leq L \leq 2^\lambda - 1$, *if* $v_L > 0$, *then for any* $0 \leq \Gamma \leq 2^\lambda - 1$ *such that* $\Gamma \sqsubseteq L$, $v_\Gamma > 0$.
2. *For any set of* $r (3 \leq r \leq \lambda)$ *numbers* $0 \leq l_0 < \cdots < l_{r-1} \leq \lambda - 1$, *if it satisfies the condition that for any* $0 \leq i < j \leq r - 1$, $v_{(2^{l_i}+2^{l_j})} > 0$, *then for the number* $L = \sum_{i=0}^{r-1} 2^{l_i}$, $v_L > 0$.
3. *The system of equations on unknowns* $\tilde{z}_\Gamma$*'s (for all* $0 \leq \Gamma \leq 2^\lambda - 1$*) and* $\tilde{w}_{\Gamma,i}$*'s (for all* $\Gamma \in M$ *and* $0 \leq i \leq |Y_\Gamma| - 1$*)*

$$\sum_{0 \leq \Gamma \leq 2^\lambda - 1: \Gamma \sqsupseteq L} \tilde{z}_\Gamma = k_L, \quad \text{for all } L \in P \tag{13}$$

$$\sum_{i=0}^{|Y_\Gamma|-1} \tilde{w}_{\Gamma,i} \leq \tilde{z}_\Gamma, \quad \text{for all } \Gamma \in M \tag{14}$$

$$\sum_{\substack{\Gamma \in M, 0 \leq i \leq |Y_\Gamma|-1: \\ \delta_{\Gamma,i} \in \rho_L}} \tilde{w}_{\Gamma,i} \geq 1, \quad \text{for all } L \in Z_2 \tag{15}$$

has a non-negative integer solution. $\square$

Proof. "**only if**" part: Statement 1 in the theorem is due to Theorem 4 and Statement 2 in the theorem is due to Theorem 5.

Since D satisfies the given intersection pattern, then by Lemma 5, there exists another matrix D' which also satisfies the given intersection pattern and each column of which is in the set F. For any $0 \le \Gamma \le 2^\lambda - 1$, let $\tilde{z}_\Gamma = z_\Gamma$, where z_Γ's are defined on the root cube-variable matrix $t(D')$ according to Definition 6. For any $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$, let $\tilde{w}_{\Gamma,i} = w_{\Gamma,i}$, where $w_{\Gamma,i}$'s are defined on the matrix D' according to Definition 17. By Theorems 6 and 7, the set of numbers $\tilde{z}_\Gamma$ and $\tilde{w}_{\Gamma,i}$ satisfies the system of equations (13)–(15). Since $\tilde{z}_\Gamma$ is the number of occurrences of the column pattern ψ_Γ in the root cube-variable matrix $t(D')$ and $\tilde{w}_{\Gamma,i}$ is the number of occurrences of the column pattern $\delta_{\Gamma,i}$ in the matrix D', therefore, $\tilde{z}_\Gamma$'s and $\tilde{w}_{\Gamma,i}$'s are all non-negative integers. Thus, the system of equations (13)–(15) has a non-negative integer solution.

"**if**" part: Let a non-negative integer solution to the system of equations (13)–(15) be $\tilde{z}_\Gamma = z_\Gamma$, for all $0 \le \Gamma \le 2^\lambda - 1$, and $\tilde{w}_{\Gamma,i} = w_{\Gamma,i}$, for all $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$. Since for all $0 \le \Gamma \le 2^\lambda - 1$, $z_\Gamma \ge 0$, for all $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$, $w_{\Gamma,i} \ge 0$, and for all $\Gamma \in M$, $\sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i} \le z_\Gamma$, then, we can construct a cube-variable matrix D as follows:

1. For all $\Gamma \in \overline{M}$, the matrix contains z_Γ columns of the form ψ_Γ.
2. For all $\Gamma \in M$, the matrix contains $\left(z_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i}\right)$ columns of the form ψ_Γ.
3. For all $\Gamma \in M$ and all $0 \le i \le |Y_\Gamma| - 1$, the matrix contains $w_{\Gamma,i}$ columns of the form $\delta_{\Gamma,i}$.

All columns of the matrix D are in the set F. Next, we prove that the matrix D satisfies the given intersection pattern, i.e., for all $0 \le \Gamma \le 2^\lambda - 1$, $V(C^\Gamma) = v_\Gamma$.

We first show that for any $L \in Z_2$, $C^L = 0$ and for any $L \in P_2$, $C^L \ne 0$. For any $L \in Z_2$, suppose that $L = 2^i + 2^j$, where $0 \le i < j \le \lambda - 1$. Since

$$\sum_{\substack{\Gamma \in M, 0 \le k \le |Y_\Gamma|-1: \\ \delta_{\Gamma,k} \in \rho_L}} w_{\Gamma,k} \ge 1,$$

there exists a $\Gamma^* \in M$ and a $0 \le k^* \le |Y_{\Gamma^*}| - 1$, such that $\delta_{\Gamma^*,k^*} \in \rho_L$ and $w_{\Gamma^*,k^*} \ge 1$. Therefore, the matrix D contains a column vector W which is from the set ρ_L. Based on the definition of ρ_L, $W_i = 0$ and $W_j = 1$, or $W_i = 1$ and $W_j = 0$. Thus, we have $C^L = c_i \cdot c_j = 0$. Thus, for any $L \in Z_2$, $C^L = 0$.

Now consider any $L \in P_2$. Suppose that $L = 2^i + 2^j$, where $0 \le i < j \le \lambda - 1$. We argue that $C^L = c_i \cdot c_j \ne 0$. Otherwise, $c_i \cdot c_j = 0$. Therefore, there exists a column r in D, such $D_{ir} = 0$ and $D_{jr} = 1$ or $D_{ir} = 1$ and $D_{jr} = 0$. Since all the columns of D are in the set F, thus the column $D_{\cdot r}$ must be in the set Y. However, based on the definition of the representative compatible column pattern set, each element W in the set Y satisfies that for the $L \in P_2$, the situation that $W_i = 0$ and $W_j = 1$ or $W_i = 1$ and $W_j = 0$ does not happen. Therefore, the column $D_{\cdot r}$ does not belong to the set Y. We get a contradiction. Thus, for any $L \in P_2$, we have $C^L \ne 0$.

Note that the given intersection pattern satisfies the conditions of Lemma 8 and the set of cubes obtained from the matrix D satisfies the condition that for any $\Gamma \in Z_2$, $C^\Gamma = 0$ and for any $\Gamma \in P_2$, $C^\Gamma \ne 0$. Therefore, based on Lemma 8, the set of cubes satisfies the condition that for any $\Gamma \in Z$, $C^\Gamma = 0$ and for any $\Gamma \in P$, $C^\Gamma \ne 0$. Thus, for all these $\Gamma \in Z$, $V(C^\Gamma) = 0 = v_\Gamma$.

Next, we will prove that for all $L \in P$, $V(C^L) = v_L$. When $L = 0$, we have $V(C^0) = 2^n = v_0$.

For any $L \in P$ and $L > 0$, L can be represented as $L = \sum_{j=0}^{r-1} 2^{l_j}$, where $1 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$. The number of $*$'s in the cube-variable row vector of C^L is the number of columns in D, whose entries on the rows $l_0, l_1, \ldots, l_{r-1}$ are all $*$'s. Due to our construction, each column of the matrix D is either of the form ψ_Γ or of the form $\delta_{\Gamma,i}$. Based on Definitions 5 and 7, a column pattern ψ_Γ has all entries on the rows $l_0, l_1, \ldots, l_{r-1}$ being $*$'s if and only if $\Gamma \sqsupseteq L$. Since the root column vector of $\delta_{\Gamma,i}$ is ψ_Γ, thus for any $\Gamma \in M$ and any $0 \le i \le |Y_\Gamma| - 1$, the column pattern $\delta_{\Gamma,i}$ has all entries on the rows $l_0, l_1, \ldots, l_{r-1}$ being $*$'s if and only if $\Gamma \sqsupseteq L$. Therefore, the number of columns in D that has $*$'s on the rows $l_0, l_1, \ldots, l_{r-1}$ is

$$\sum_{\substack{\Gamma \in \overline{M}: \\ \Gamma \sqsupseteq L}} z_\Gamma + \sum_{\substack{\Gamma \in M: \\ \Gamma \sqsupseteq L}} \left(z_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i} \right) + \sum_{\substack{\Gamma \in M: \\ \Gamma \sqsupseteq L}} \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i} = \sum_{0 \le \Gamma \le 2^\lambda - 1: \Gamma \sqsupseteq L} z_\Gamma.$$

Since z_Γ's let Eq. (13) hold, the right-hand side of the above equation is equal to k_L. Therefore, the number of $*$'s in the cube-variable row vector of C^L is k_L. Since C^L is nonempty, by Lemma 1, $V(C^L) = 2^{k_L}$. Thus, for any $L \in P$ and $L > 0$, $V(C^L) = 2^{k_L} = v_L$.

In summary, for any $0 \le \Gamma \le 2^\lambda - 1$, $V(C^\Gamma) = v_\Gamma$. Thus, the matrix D satisfies the given intersection pattern. □

Note that when all the three statements in the above theorem hold, the above proof provides a way to synthesize a cube-variable matrix to satisfy the given intersection pattern. Indeed, suppose that a non-negative integer solution to the system of equations (13)–(15) is $\tilde{z}_\Gamma = z_\Gamma$, for all $0 \le \Gamma \le 2^\lambda - 1$, and $\tilde{w}_{\Gamma,i} = w_{\Gamma,i}$, for all $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$. Then, we can construct a cube-variable matrix that satisfies the given intersection pattern as follows:

1. For all $\Gamma \in \overline{M}$, the matrix contains z_Γ columns of the form ψ_Γ.
2. For all $\Gamma \in M$, the matrix contains $\left(z_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i}\right)$ columns of the form ψ_Γ.
3. For all $\Gamma \in M$ and all $0 \le i \le |Y_\Gamma| - 1$, the matrix contains $w_{\Gamma,i}$ columns of the form $\delta_{\Gamma,i}$.

Example 11. Given $v_0 = 64, v_1 = 4, v_2 = 8, v_3 = 0, v_4 = 16, v_5 = 2, v_6 = 2, v_7 = 0, v_8 = 8, v_9 = 1, v_{10} = 2, v_{11} = 0, v_{12} = 0, v_{13} = 0, v_{14} = 0, v_{15} = 0$, determine whether there exists a set of four cubes $c_0, \ldots, c_3$ on 6 variables $x_0, \ldots, x_5$ that satisfies the intersection pattern $(v_0, \ldots, v_{15})$.

Solution: First, it is not hard to check that both Statement 1 and Statement 2 in Theorem 8 hold for the given pattern.

Now we check whether Statement 3 in Theorem 8 holds. For the given intersection pattern, we have $P = \{0, 1, 2, 4, 5, 6, 8, 9, 10\}$, $Z = \{3, 7, 11, 12, 13, 14, 15\}$, and

$$\begin{array}{lllll} k_0 = 6, & k_1 = 2, & k_2 = 3, & k_4 = 4, & k_5 = 1, \\ k_6 = 1, & k_8 = 3, & k_9 = 0, & k_{10} = 1. & \end{array}$$

Notice that $Z_2 = \{3, 12\}$. The corresponding representative compatible column pattern sets are $\rho_3 = \{(0, 1, *, *)^T\}$ and $\rho_{12} = \{(*, *, 0, 1)^T\}$, respectively. Thus, we have

$$Y = \bigcup_{\Gamma \in Z_2} \rho_\Gamma = \{(0, 1, *, *)^T, (*, *, 0, 1)^T\}.$$

Since the root column vector of $(0, 1, *, *)^T$ is ψ_{12} and the root column vector of $(*, *, 0, 1)^T$ is ψ_3, we have $M = \{3, 12\}$. We can partition the set Y as $Y_3 = \{(*, *, 0, 1)^T\}$ and $Y_{12} = \{(0, 1, *, *)^T\}$.

Based on Definition 17, the element in the set Y_3 is defined as $\delta_{3,0} = (*, *, 0, 1)^T$ and the element in the set Y_{12} is defined as $\delta_{12,0} = (0, 1, *, *)^T$. Notice that $\rho_3 = \{\delta_{12,0}\}$ and $\rho_{12} = \{\delta_{3,0}\}$.

We can derive the system of equations (13)–(15) for this example as

$$\begin{cases} \sum_{i=0}^{15} \tilde{z}_i = 6 \\ \tilde{z}_1 + \tilde{z}_3 + \tilde{z}_5 + \tilde{z}_7 + \tilde{z}_9 + \tilde{z}_{11} + \tilde{z}_{13} + \tilde{z}_{15} = 2 \\ \tilde{z}_2 + \tilde{z}_3 + \tilde{z}_6 + \tilde{z}_7 + \tilde{z}_{10} + \tilde{z}_{11} + \tilde{z}_{14} + \tilde{z}_{15} = 3 \\ \tilde{z}_4 + \tilde{z}_5 + \tilde{z}_6 + \tilde{z}_7 + \tilde{z}_{12} + \tilde{z}_{13} + \tilde{z}_{14} + \tilde{z}_{15} = 4 \\ \tilde{z}_5 + \tilde{z}_7 + \tilde{z}_{13} + \tilde{z}_{15} = 1 \\ \tilde{z}_6 + \tilde{z}_7 + \tilde{z}_{14} + \tilde{z}_{15} = 1 \\ \sum_{i=8}^{15} \tilde{z}_i = 3 \\ \tilde{z}_9 + \tilde{z}_{11} + \tilde{z}_{13} + \tilde{z}_{15} = 0 \\ \tilde{z}_{10} + \tilde{z}_{11} + \tilde{z}_{14} + \tilde{z}_{15} = 1 \\ \tilde{w}_{3,0} \le \tilde{z}_3 \\ \tilde{w}_{12,0} \le \tilde{z}_{12} \\ \tilde{w}_{3,0} \ge 1 \\ \tilde{w}_{12,0} \ge 1. \end{cases}$$

Note that the first 9 equations correspond to Eq. (13), the next 2 equations correspond to Eq. (14), and the last 2 equations correspond to Eq. (15).

The above system of equations has a non-negative solution

$$\begin{array}{l} \tilde{z}_3 = 1, \quad \tilde{z}_4 = 1, \quad \tilde{z}_7 = 1, \quad \tilde{z}_{10} = 1, \quad \tilde{z}_{12} = 2, \\ \tilde{z}_0 = \tilde{z}_1 = \tilde{z}_2 = \tilde{z}_5 = \tilde{z}_6 = \tilde{z}_8 = \tilde{z}_9 = \tilde{z}_{11} = \tilde{z}_{13} = \tilde{z}_{14} = \tilde{z}_{15} = 0, \\ \tilde{w}_{3,0} = 1, \quad \tilde{w}_{12,0} = 1. \end{array}$$

Thus, Statement 3 in Theorem 8 also holds. Therefore, there exists a cube-variable matrix that satisfies the given intersection pattern. We can construct a cube-variable matrix that satisfies the given intersection pattern based on the above non-negative solution as follows:

1. For all $\Gamma \in \overline{M}$, the matrix contains $\tilde{z}_\Gamma$ columns of the form ψ_Γ. Since $M = \{3, 12\}$, we have $\overline{M} = \{0, 1, 2, 4, 5, \ldots, 11, 13, 14, 15\}$. Thus, the matrix contains one column of the pattern $\psi_4 = (1, 1, *, 1)^T$, one column of the pattern $\psi_7 = (*, *, *, 1)^T$, and one column of the pattern $\psi_{10} = (1, *, 1, *)^T$.
2. For all $\Gamma \in M$, the matrix contains $\left(\tilde{z}_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} \tilde{w}_{\Gamma,i}\right)$ columns of the form ψ_Γ. In this example, $M = \{3, 12\}$. Based on the non-negative solution, we have that for $\Gamma = 3$,

$$\tilde{z}_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} \tilde{w}_{\Gamma,i} = \tilde{z}_3 - \tilde{w}_{3,0} = 0;$$

for $\Gamma = 12$,

$$\tilde{z}_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} \tilde{w}_{\Gamma,i} = \tilde{z}_{12} - \tilde{w}_{12,0} = 1.$$

Therefore, the matrix contains one column of the pattern $\psi_{12} = (1, 1, *, *)^T$.

3. For all $\Gamma \in M$ and all $0 \le i \le |Y_\Gamma| - 1$, the matrix contains $\tilde{w}_{\Gamma,i}$ columns of the form $\delta_{\Gamma,i}$. In this example, $M = \{3, 12\}$. Based on the non-negative solution, the matrix contains one column of the pattern $\delta_{3,0} = (*, *, 0, 1)^T$ and one column of the pattern $\delta_{12,0} = (0, 1, *, *)^T$.

Consequently, a matrix that satisfies the given intersection pattern is

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

and the corresponding cubes are

$$c_0 = x_0x_2x_3\bar{x}_5, \qquad c_1 = x_0x_3x_5, \qquad c_2 = x_2\bar{x}_4, \qquad c_3 = x_0x_1x_4.$$

It is not hard to verify that the set of cubes $c_0, \dots, c_3$ satisfies the given intersection pattern. □

5. Implementation

In this section, we will discuss the implementation of the procedure to solve the λ-cube intersection problem, based on the theoretical results in Section 4.

5.1. Checking Statement 1 in Theorem 8

We can represent Statement 1 in Theorem 8 in an alternative way, as shown by the following theorem.

Theorem 9. *The following two statements are equivalent:*

1. *The intersection pattern $(v_0, \dots, v_{2^\lambda-1})$ satisfies the condition that for any number $0 \le L \le 2^\lambda - 1$, if $v_L > 0$, then for any number $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq L$, $v_\Gamma > 0$.*
2. *The intersection pattern $(v_0, \dots, v_{2^\lambda-1})$ satisfies the condition that for any $1 \le k \le \lambda$ and any number $L \in P_k$, if a number $0 \le \Gamma \le 2^\lambda - 1$ satisfies that $\|\Gamma\| = k - 1$ and $\Gamma \sqsubseteq L$, then $v_\Gamma > 0$. (Note that the operator $\|\cdot\|$ and the set P_k are defined in Definition 10.)* □

Proof. Statement 1 ⇒ Statement 2: Consider any $L \in P_k$, where $1 \le k \le \lambda$. By the definition of P_k, we have $v_L > 0$. Since Statement 1 holds, therefore, for any $0 \le \Gamma \le 2^\lambda - 1$ such that $\|\Gamma\| = k - 1$ and $\Gamma \sqsubseteq L$, we have $v_\Gamma > 0$. Thus, Statement 2 holds.

Statement 2 ⇒ Statement 1: When $L = 0$, we have $v_0 = 2^n > 0$. Notice that the only $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq 0$ is $\Gamma = 0$. Thus, for any $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq 0$, we have $v_\Gamma > 0$.

Now consider any $1 \le L \le 2^\lambda - 1$ such that $v_L > 0$. Suppose that $\|L\| = r$. Then, $1 \le r \le \lambda$ and $L \in P_r$. For any Γ such that $0 \le \Gamma \le 2^\lambda - 1$ and $\Gamma \sqsubseteq L$, suppose that $\|\Gamma\| = t$. Then, we have $0 \le t \le r$. We can find $r - t + 1$ numbers $\Gamma_t, \dots, \Gamma_r$, such that $\Gamma_t = \Gamma$, $\Gamma_r = L$, and for any $t \le k \le r - 1$, $\|\Gamma_k\| = k$ and $\Gamma_k \sqsubseteq \Gamma_{k+1}$. Since Statement 2 holds and $v_{\Gamma_r} = v_L > 0$, we can see that for any $t \le k \le r - 1$, $v_{\Gamma_k} > 0$. In particular, $v_\Gamma = v_{\Gamma_t} > 0$. Thus, for any $0 \le \Gamma \le 2^\lambda - 1$ such that $\Gamma \sqsubseteq L$, we have $v_\Gamma > 0$. This concludes the proof. □

Based on Theorem 9, in order to check whether Statement 1 in Theorem 8 holds, we only need to check whether Statement 2 in Theorem 9 holds. Thus, whether Statement 1 in Theorem 8 holds can be checked by a procedure shown in Algorithm 1. The procedure begins by obtaining the sets $P_0, P_1, \dots, P_\lambda$ from the intersection pattern $(v_0, \dots, v_{2^\lambda-1})$ (Lines 2–6). Then, starting from the set P_1 and ending at the set P_λ, the procedure will check whether each number L in the set P_i satisfies the condition that for any number $0 \le \Gamma \le 2^\lambda - 1$ such that $\|\Gamma\| = i - 1$ and $\Gamma \sqsubseteq L$, we have $v_\Gamma > 0$ (Lines 7–12).

The time complexity of obtaining the sets $P_0, P_1, \dots, P_\lambda$ (Lines 2–6) is

$$T_1 = O(\lambda 2^\lambda),$$

because obtaining $\|\Gamma\|$ for each number Γ takes $O(\lambda)$ time units and we need to perform that operation 2^λ times.

The time complexity of checking Statement 2 in Theorem 9 (Lines 7–12) can be analyzed as follows. For each $L \in P_i$, we need to obtain those Γ such that $\Gamma \sqsubseteq L$ and $\|\Gamma\| = i - 1$, and check whether v_Γ is positive or not (Lines 9–11). Given an $L \in P_i$, there are i numbers Γ satisfying that $\Gamma \sqsubseteq L$ and $\|\Gamma\| = i - 1$. They can be obtained by replacing one "1" in the binary

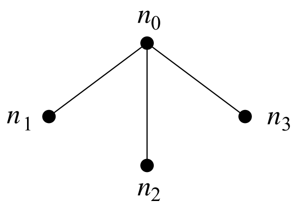

Fig. 3. An undirected graph constructed from the intersection pattern of Example 5.

representation of L by a "0". Therefore, the time complexity of Lines 9–11 is $O(i)$. The total time complexity for checking Statement 2 in Theorem 9 is

$$T_2 = \sum_{i=1}^{\lambda} |P_i| \cdot O(i),$$

where $|P_i|$ is the cardinality of the set P_i. Based on the definition of P_i, we can see that the maximum number of elements in P_i is bounded by the number of values with exactly i ones in their binary representations. Thus,

$$|P_i| \le \binom{\lambda}{i}.$$

Therefore, the total time complexity for checking the statement is

$$T_2 = \sum_{i=1}^{\lambda} |P_i| \cdot O(i) \le O\left(\sum_{i=1}^{\lambda} i\binom{\lambda}{i}\right) = O\left(\sum_{i=1}^{\lambda} \lambda\binom{\lambda-1}{i-1}\right) = O\left(\lambda 2^{\lambda-1}\right) = O\left(\lambda 2^{\lambda}\right).$$

The total time complexity of Algorithm 1 is

$$T_1 + T_2 = O(\lambda 2^{\lambda}).$$

Note that the input to the λ-cube intersection problem consists of $N = 2^{\lambda}$ numbers $v_0, \ldots, v_{2^{\lambda}-1}$. Thus, in terms of the input size, the time complexity of Algorithm 1 is

$$O(N \log_2 N).$$

Algorithm 1 CheckRuleOne(λ, v): the procedure to check whether Statement 1 in Theorem 8 holds. It returns 1 if the statement holds; otherwise, it returns 0.

1: {Given an integer $\lambda \ge 1$ and a non-negative integer array $v = (v_0, \ldots, v_{2^{\lambda}-1})$.}
2: **for** $i \Leftarrow 0$ to λ **do**
3:   $P_i \Leftarrow \phi$;
4: **for** $\Gamma \Leftarrow 0$ to $2^{\lambda} - 1$ **do**
5:   **if** $v_{\Gamma} > 0$ **then**
6:     $k \Leftarrow ||\Gamma||$; $P_k \Leftarrow P_k \cup \{\Gamma\}$;
7: **for** $i \Leftarrow 1$ to λ **do**
8:   **for all** $L \in P_i$ **do**
9:     **for all** $0 \le \Gamma \le 2^{\lambda} - 1$ s.t. $||\Gamma|| = i - 1$ and $\Gamma \sqsubseteq L$ **do**
10:       **if** $v_{\Gamma} = 0$ **then**
11:         **return** 0;
12: **return** 1;

5.2. Checking Statement 2 in Theorem 8

Whether Statement 2 in Theorem 8 holds can be checked by representing the given intersection pattern by an undirected graph and listing all maximal cliques of the undirected graph.

For a given intersection pattern on λ cubes, we can construct an undirected graph $G(N, E)$ from that pattern, where N is a set of λ nodes $n_0, \ldots, n_{\lambda-1}$ and E is a set of edges. There is an edge between the nodes n_i and n_j $(0 \le i < j \le \lambda - 1)$ if and only if the number $(2^i + 2^j)$ is in the set P_2.

For example, we can represent the intersection pattern shown in Example 5 by the undirected graph shown in Fig. 3.

In graph theory, a *clique* in an undirected graph $G(N, E)$ is defined as a subset Q of the node set N, such that for every two nodes in Q, there exists an edge connecting the two. A *maximal clique* is a clique that cannot be extended by including one more adjacent node.

For an intersection pattern, if a set of r $(3 \le r \le \lambda)$ numbers $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$ satisfies that for any $0 \le i < j \le r - 1$, $v_{(2^{l_i}+2^{l_j})} > 0$, then, the set of nodes $n_{l_0}, \ldots, n_{l_{r-1}}$ forms a clique of the undirected graph constructed from the intersection pattern. Thus, Statement 2 in Theorem 8 can be stated in another way as: For any clique $Q = \{n_{l_0}, \ldots, n_{l_{r-1}}\}$ of size r in the undirected graph constructed from the intersection pattern, where $3 \le r \le \lambda$ and $0 \le l_0 < \cdots < l_{r-1} \le \lambda - 1$, we have $v_L > 0$, where $L = \sum_{i=0}^{r-1} 2^{l_i}$.

The following theorem shows that if Statement 1 in Theorem 8 holds, then to check whether Statement 2 holds, we only need to focus on all maximal cliques of the undirected graph $G(N, E)$.

Theorem 10. *If Statement 1 in Theorem 8 holds, then Statement 2 in Theorem 8 holds if and only if for any maximal clique $Q^* = \{n_{d_0}, \ldots, n_{d_{t-1}}\}$ of size t ($3 \le t \le \lambda$ and $0 \le d_0 < \cdots < d_{t-1} \le \lambda - 1$) in the undirected graph constructed from the intersection pattern, we have $v_{L^*} > 0$, where $L^* = \sum_{i=0}^{t-1} 2^{d_i}$.* □

Proof. The "only if" part of the above theorem is obvious. We now prove the "if" part. Consider any clique $Q = \{n_{l_0}, \ldots, n_{l_{r-1}}\}$ in the undirected graph $G(N, E)$. By the definition of maximal clique, Q is contained in a maximal clique $Q^* = \{n_{d_0}, \ldots, n_{d_{t-1}}\}$, where $r \le t \le \lambda$, $0 \le d_0 < \cdots < d_{t-1} \le \lambda - 1$. Since the clique Q is contained in the clique Q^*, we have $Q \subseteq Q^*$. Let $L = \sum_{i=0}^{r-1} 2^{l_i}$ and $L^* = \sum_{i=0}^{t-1} 2^{d_i}$. Thus, we have $L \sqsubseteq L^*$. By our assumption, for the maximal clique Q^*, we have $v_{L^*} > 0$. Now by another assumption that Statement 1 in Theorem 8 holds, we can obtain $v_L > 0$. Thus, for any clique $Q = \{n_{l_0}, \ldots, n_{l_{r-1}}\}$ in the undirected graph $G(N, E)$, we have $v_L > 0$. Therefore, Statement 2 in Theorem 8 holds. □

Therefore, if Statement 1 in Theorem 8 holds, then whether Statement 2 in Theorem 8 holds can be answered by checking whether all v_L's corresponding to all maximal cliques in the undirected graph $G(N, E)$ are positive. The problem of listing all maximal cliques in an undirected graph is a classical problem in graph theory and can be solved, for example, by the Born–Kerbosch algorithm [2].

Assuming that Statement 1 in Theorem 8 holds, then whether Statement 2 in Theorem 8 holds can be checked by the procedure shown in Algorithm 2. The procedure begins by constructing an undirected graph based on the intersection pattern (Lines 2–6). The time complexity is $O(\lambda^2)$. Then, it obtains all maximal cliques and checks whether each v_L corresponding to each maximal clique is positive or not (Lines 7–12). The worst-case time complexity for finding all maximal cliques in a graph of λ nodes is $O(3^{\lambda/3})$ [8]. Given a maximal clique, the time complexity to obtain its corresponding L (Lines 8–10) is $O(\lambda)$. Therefore, the time complexity of Lines 7–12 is $O(\lambda 3^{\lambda/3}) = O(\lambda 2^{\lambda})$. In summary, the total time complexity for Algorithm 2 is $O(\lambda^2) + O(\lambda 2^{\lambda}) = O(\lambda 2^{\lambda}) = O(N \log_2 N)$, where N is the input size.

Algorithm 2 CheckRuleTwo(λ, v): the procedure to check whether Statement 2 in Theorem 8 holds under the assumption that Statement 1 in Theorem 8 holds. It returns 1 if the statement holds; otherwise, it returns 0.

```
1: {Given an integer λ ≥ 1 and a non-negative integer array v = (v_0, ..., v_{2^λ−1}).}
2: N ⇐ {n_0, ..., n_{λ−1}}; E ⇐ φ;
3: for i ⇐ 0 to λ − 1 do
4:     for j ⇐ i + 1 to λ − 1 do
5:         if v_(2^i+2^j) > 0 then
6:             E ⇐ E ∪ {e(n_i, n_j)}; {Add an edge between the node n_i and the node n_j into the edge set E.}
7: for all maximal clique Q in the graph G(N, E) do
8:     L ⇐ 0;
9:     for all node n_i in Q do
10:        L ⇐ L + 2^i; {Construct the number L corresponding to the maximal clique Q.}
11:    if v_L = 0 then
12:        return 0;
13: return 1;
```

5.3. Checking Statement 3 in Theorem 8

The following theorem shows that to check whether the system of equations (13)–(15) has a non-negative solution, we only need to check whether an alternative system of equations with fewer unknowns and equations has a non-negative solution.

Theorem 11. *The system of equations (13)–(15) has a non-negative integer solution if and only if the system of equations on unknowns $\hat{z}_\Gamma$'s (for all $\Gamma \in \overline{M}$) and $\hat{w}_{\Gamma,i}$'s (for all $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$)*

$$\sum_{\Gamma \in \overline{M}, \Gamma \sqsupseteq L} \hat{z}_\Gamma + \sum_{\Gamma \in M, \Gamma \sqsupseteq L} \sum_{i=0}^{|Y_\Gamma|-1} \hat{w}_{\Gamma,i} = k_L, \quad \text{for all } L \in P \tag{16}$$

$$\sum_{\substack{\Gamma \in M, 0 \le i \le |Y_\Gamma|-1: \\ \delta_{\Gamma,i} \in \rho_L}} \hat{w}_{\Gamma,i} \ge 1, \quad \text{for all } L \in Z_2 \tag{17}$$

has a non-negative integer solution. □

Proof. **"if"** part: Suppose that a non-negative integer solution to the system of equations (16) and (17) is

$$\begin{cases} \hat{z}_\Gamma = z_\Gamma, & \text{for all } \Gamma \in \overline{M}, \\ \hat{w}_{\Gamma,i} = w_{\Gamma,i}, & \text{for all } \Gamma \in M, 0 \le i \le |Y_\Gamma| - 1. \end{cases}$$

We let

$$\begin{cases} \tilde{z}_\Gamma = z_\Gamma, & \text{for all } \Gamma \in \overline{M}, \\ \tilde{z}_\Gamma = \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i}, & \text{for all } \Gamma \in M, \\ \tilde{w}_{\Gamma,i} = w_{\Gamma,i}, & \text{for all } \Gamma \in M, 0 \le i \le |Y_\Gamma| - 1. \end{cases}$$

Then, it is not hard to see that $\tilde{z}_\Gamma$'s (for all $0 \le \Gamma \le 2^\lambda - 1$) and $\tilde{w}_{\Gamma,i}$'s (for all $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$) form a non-negative integer solution to the system of equations (13)–(15).

"only if" part: Suppose that a non-negative integer solution to the system of equations (13)–(15) is

$$\begin{cases} \tilde{z}_\Gamma = z_\Gamma, & \text{for all } 0 \le \Gamma \le 2^\lambda - 1, \\ \tilde{w}_{\Gamma,i} = w_{\Gamma,i}, & \text{for all } \Gamma \in M, 0 \le i \le |Y_\Gamma| - 1. \end{cases} \tag{18}$$

We let

$$\begin{cases} \hat{z}_\Gamma = z_\Gamma, & \text{for all } \Gamma \in \overline{M}, \\ \hat{w}_{\Gamma,0} = z_\Gamma - \sum_{i=1}^{|Y_\Gamma|-1} w_{\Gamma,i}, & \text{for all } \Gamma \in M, \\ \hat{w}_{\Gamma,i} = w_{\Gamma,i}, & \text{for all } \Gamma \in M, 1 \le i \le |Y_\Gamma| - 1. \end{cases} \tag{19}$$

Then, for all $\Gamma \in \overline{M}$, $\hat{z}_\Gamma = z_\Gamma \ge 0$ and for all $\Gamma \in M$, $1 \le i \le |Y_\Gamma| - 1$, $\hat{w}_{\Gamma,i} = w_{\Gamma,i} \ge 0$. Since for all $\Gamma \in M$, $\sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i} \le z_\Gamma$, then we have that for all $\Gamma \in M$,

$$0 \le z_\Gamma - \sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i} \le z_\Gamma - \sum_{i=1}^{|Y_\Gamma|-1} w_{\Gamma,i} = \hat{w}_{\Gamma,0}.$$

Therefore, the set of numbers $\hat{z}_\Gamma$'s and $\hat{w}_{\Gamma,i}$'s given by Eq. (19) is non-negative.

Based on Eqs. (13), (18), and (19), we have that for all $L \in P$,

$$\sum_{\Gamma \in \overline{M}, \Gamma \sqsupseteq L} \hat{z}_\Gamma + \sum_{\Gamma \in M, \Gamma \sqsupseteq L} \sum_{i=0}^{|Y_\Gamma|-1} \hat{w}_{\Gamma,i} = \sum_{\Gamma \in \overline{M}, \Gamma \sqsupseteq L} z_\Gamma + \sum_{\Gamma \in M, \Gamma \sqsupseteq L} z_\Gamma = \sum_{0 \le \Gamma \le 2^\lambda - 1, \Gamma \sqsupseteq L} \tilde{z}_\Gamma = k_L. \tag{20}$$

Since for all $\Gamma \in M$, $\sum_{i=0}^{|Y_\Gamma|-1} w_{\Gamma,i} \le z_\Gamma$, then we have that for all $\Gamma \in M$,

$$\hat{w}_{\Gamma,0} = z_\Gamma - \sum_{i=1}^{|Y_\Gamma|-1} w_{\Gamma,i} \ge w_{\Gamma,0}. \tag{21}$$

Based on Eqs. (21) and (19), we have that for all $\Gamma \in M$ and $0 \le i \le |Y_\Gamma| - 1$,

$$\hat{w}_{\Gamma,i} \ge w_{\Gamma,i}. \tag{22}$$

Combining Eq. (22) with Eqs. (15) and (18), we have that for all $L \in Z_2$,

$$1 \le \sum_{\substack{\Gamma \in M, 0 \le i \le |Y_\Gamma|-1: \\ \delta_{\Gamma,i} \in \rho_L}} \tilde{w}_{\Gamma,i} = \sum_{\substack{\Gamma \in M, 0 \le i \le |Y_\Gamma|-1: \\ \delta_{\Gamma,i} \in \rho_L}} w_{\Gamma,i} \le \sum_{\substack{\Gamma \in M, 0 \le i \le |Y_\Gamma|-1: \\ \delta_{\Gamma,i} \in \rho_L}} \hat{w}_{\Gamma,i}. \tag{23}$$

Since both Eqs. (20) and (23) hold, we conclude that $\hat{z}_\Gamma$(for all $\Gamma \in \overline{M}$) and $\hat{w}_{\Gamma,i}$(for all $\Gamma \in M$, $1 \le i \le |Y_\Gamma| - 1$) form a non-negative integer solution to the system of equations (16) and (17). □

Based on Theorem 11, we can check whether Statement 3 in Theorem 8 holds by checking whether the system of equations (16) and (17) has a non-negative solution. Note that the system of equations (16) and (17) has $|M|$ fewer unknowns and $|M|$ fewer inequalities than the original system of equations (13)–(15). Experimental results in Section 6 on a number of benchmarks showed that on average the system of equations (16) and (17) has 17.1% fewer unknowns and 58.5% fewer inequalities than the system of equations (13)–(15).

The system of equations (16) and (17) is a set of linear equations and inequalities. Thus, it can be represented in matrix form as

$$\begin{cases} A_{ze}\vec{z} + A_{we}\vec{w} = b_e, \\ A_w\vec{w} \geq b, \end{cases} \tag{24}$$

where A_{ze} and A_{we} are (0, 1)-matrices obtained from Eq. (16), b_e is a column vector of k_L's, A_w is a (0, 1)-matrix obtained from Eq. (17), b is a column vector of ones, $\vec{z}$ is a column vector of unknowns $\hat{z}_\Gamma$'s, for all $\Gamma \in \overline{M}$, and $\vec{w}$ is a column vector of unknowns $\hat{w}_{\Gamma,i}$'s, for all $\Gamma \in M$ and $0 \leq i \leq |Y_\Gamma| - 1$.

Note that when the necessary and sufficient condition listed in Theorem 8 is satisfied, then we can construct a cube-variable matrix that satisfies the given intersection pattern based on a non-negative integer solution to the system of equations (16) and (17). Indeed, suppose that a non-negative integer solution is $\hat{z}_\Gamma = z_\Gamma$, for all $\Gamma \in \overline{M}$, and $\hat{w}_{\Gamma,i} = w_{\Gamma,i}$, for all $\Gamma \in M$ and $0 \leq i \leq |Y_\Gamma| - 1$. Then, we can construct a cube-variable matrix that satisfies the given intersection pattern as follows:

1. For all $\Gamma \in \overline{M}$, the matrix contains z_Γ columns of the form ψ_Γ.
2. For all $\Gamma \in M$ and all $0 \leq i \leq |Y_\Gamma| - 1$, the matrix contains $w_{\Gamma,i}$ columns of the form $\delta_{\Gamma,i}$.

5.4. The procedure for solving the λ-cube intersection problem

Based on the above discussion, we give the procedure for solving the λ-cube intersection problem in Algorithm 3. In the procedure, the function CheckRuleOne(λ, v) and the function CheckRuleTwo(λ, v) are shown in Algorithm 1 and 2, respectively. The function RCCPS(Γ, λ, P_2) returns the representative compatible column pattern set for a $\Gamma \in Z_2$. The function $t(W)$ is defined in Definition 14, which returns the root column vector of a column W. The function GetIndex(W) takes a column $W \in \Psi$ and returns the index Γ such that $W = \psi_\Gamma$. The function

$$\text{SetEqn}(P, Z_2, M, \overline{M}, \{k_L|L \in P\}, \{\rho_L|L \in Z_2\}, \{Y_L|L \in M\})$$

returns the matrices A_{ze}, A_{we}, and A_w and the column vectors b_e and b shown in Eq. (24). The function NonNegSln($A_{ze}, A_{we}, b_e, A_w, b$) finds a non-negative integer solution to Eq. (24). If Eq. (24) has a non-negative integer solution, then the function returns one such solution; otherwise, it returns ϕ. Given a non-negative solution $(\vec{z}, \vec{w})$ to Eq. (24), the function SynCubes($\vec{z}, \vec{w}, \lambda$) synthesizes a set of λ cubes that satisfies the given intersection pattern.

Algorithm 3 CubePattern(λ, v): the procedure to check whether there exists a set of λ cubes that satisfies the given intersection pattern $v = (v_0, \ldots, v_{2^\lambda-1})$. If the answer is yes, the procedure returns a set of cubes that satisfies the intersection pattern; otherwise, it returns ϕ.

{Given an integer $\lambda \geq 1$ and a non-negative integer array $v = (v_0, \ldots, v_{2^\lambda-1})$, where each entry is from the set $\{0, 2^0, 2^1, \ldots, 2^n\}$.}
if CheckRuleOne(λ, v) $= 0$ **then return** ϕ;
if CheckRuleTwo(λ, v) $= 0$ **then return** ϕ;
$P \Leftarrow \phi$; $P_2 \Leftarrow \phi$; $Z_2 \Leftarrow \phi$; $Y \Leftarrow \phi$; $M \Leftarrow \phi$; $\overline{M} \Leftarrow \phi$;
for $i \Leftarrow 0$ to $2^\lambda - 1$ **do** {Obtain the set P and the values k_Γ's.}
  if $v_\Gamma > 0$ **then** $P \Leftarrow P \cup \{\Gamma\}$; $k_\Gamma \Leftarrow \log_2 v_\Gamma$;
for $i \Leftarrow 0$ to $\lambda - 1$ **do** {Obtain the sets P_2 and Z_2.}
  for $j \Leftarrow i + 1$ to $\lambda - 1$ **do**
    if $v_{(2^i+2^j)} > 0$ **then** $P_2 \Leftarrow P_2 \cup \{2^i + 2^j\}$;
    else $Z_2 \Leftarrow Z_2 \cup \{2^i + 2^j\}$;
for all $\Gamma \in Z_2$ **do** {Obtain the sets ρ_Γ's and Y.}
  $\rho_\Gamma \Leftarrow$ RCCPS(Γ, λ, P_2); $Y \Leftarrow Y \cup \rho_\Gamma$;
for all $W \in Y$ **do** {Obtain the set M.}
  $\Gamma \Leftarrow$ GetIndex($t(W)$); $M \Leftarrow M \cup \{\Gamma\}$;
for $\Gamma \Leftarrow 0$ to $2^\lambda - 1$ **do** {Obtain the set $\overline{M}$.}
  if $\Gamma \notin M$ **then** $\overline{M} \Leftarrow \overline{M} \cup \{\Gamma\}$;
for all $\Gamma \in M$ **do**
  $Y_\Gamma \Leftarrow \phi$;
for all $W \in Y$ **do** {Obtain the sets Y_Γ's.}
  $\Gamma \Leftarrow$ GetIndex($t(W)$); $Y_\Gamma \Leftarrow Y_\Gamma \cup \{W\}$;
$(A_{ze}, A_{we}, b_e, A_w, b) \Leftarrow$ SetEqn($P, Z_2, M, \overline{M}, \{k_L|L \in P\}, \{\rho_L|L \in Z_2\}, \{Y_L|L \in M\}$);
$(\vec{z}, \vec{w}) \Leftarrow$ NonNegSln($A_{ze}, A_{we}, b_e, A_w, b$);
if $(\vec{z}, \vec{w}) = \phi$ **then**
  return ϕ;
return SynCubes($\vec{z}, \vec{w}, \lambda$);

5.5. Time complexity analysis of Algorithm 3

In this section, we analyze the time complexity of Algorithm 3, the procedure to solve the λ-cube intersection problem.

As we have shown in Sections 5.1 and 5.2, the time complexities for the functions CheckRuleOne(λ, v) and CheckRuleTwo(λ, v) are both $O(\lambda 2^\lambda)$.

The time complexity of obtaining the set P and the values k_Γ's (Lines 5–6) is $O(2^\lambda)$. The time complexity of obtaining the sets P_2 and Z_2 (Lines 7–10) is $O(\lambda^2)$.

Now we analyze the time complexity of obtaining all the sets ρ_Γ's for all $\Gamma \in Z_2$ (Lines 11–12). Since each element in a representative compatible column pattern set ρ_Γ for a $\Gamma \in Z_2$ is a length-λ vector composed of either 0, 1, or $*$, the size of the set ρ_Γ is bounded by 3^λ. The time complexity of obtaining such a set is $O(3^\lambda)$. Thus, the time complexity of obtaining all the sets ρ_Γ's for all $\Gamma \in Z_2$ (Lines 11–12) is $O(\lambda^2 3^\lambda)$. Indeed, the worst case happens when $P_2 = \phi$ and $Z_2 = \{\Gamma | 0 \le \Gamma \le 2^\lambda - 1, \|\Gamma\| = 2\}$. In this case, we need to obtain $O(\lambda^2)$ sets ρ_Γ's; obtaining each set takes $O(3^\lambda)$ time units.

Since each element in the set Y is a length-λ vector composed of either 0, 1, or $*$, the size of the set Y is bounded by 3^λ. For each element in the set Y, the time complexity of obtaining its root column vector and the index Γ is $O(\lambda)$. Thus, the time complexity of obtaining the set M (Lines 13–14) is $O(\lambda 3^\lambda)$. Similarly, we can show that the time complexity of obtaining the sets Y_Γ's for all $\Gamma \in M$ (Lines 17–20) is $O(\lambda 3^\lambda)$.

The time complexity of obtaining the set $\overline{M}$ (Lines 15–16) is $O(2^\lambda)$.

Now we analyze the time complexity of establishing Eq. (24) (Line 21). Note that the number of variables is equal to

$$|\overline{M}| + \sum_{\Gamma \in M} |Y_\Gamma| = |\overline{M}| + |Y| = O(2^\lambda) + O(3^\lambda) = O(3^\lambda).$$

The number of equations and inequalities in Eq. (24) is equal to $|P| + |Z_2|$. Note that

$$|P| + |Z_2| \le |P| + |Z| = 2^\lambda.$$

Thus, the sum of the numbers of entries in the matrices A_{ze}, A_{we}, and A_w is bounded by $2^\lambda O(3^\lambda) = O(6^\lambda)$.

Putting all the above analysis together, we conclude that without considering the time complexity in solving Eq. (24) (Line 22), the total time complexity of Algorithm 3 is $O(6^\lambda)$. Since the input size of the λ-cube intersection problem is $N = 2^\lambda$, the total time complexity of Algorithm 3 (without considering the time complexity in solving Eq. (24)) is $O(N^{\log_2 6})$.

6. Experimental results

We tested our algorithm on two-level logic circuit benchmarks that accompany the two-level logic minimizer Espresso [1]. For each benchmark, we ignored the output part of the cubes and just set the number of outputs to one. We optimized each modified benchmark by Espresso and then generated an intersection pattern for the set of cubes in that benchmark. This intersection pattern serves as the input to our program.[2]

We performed two sets of experiments to test our algorithm. In the first set of experiments, we tested our algorithm on solving special cases. The main goal was to study the runtime of our algorithm. The benchmarks we tested are listed in Table 1. Since just a few benchmarks generate a special intersection pattern with $v_{2^\lambda-1} > 0$, we manually created some test cases from the existing ones. For example, the intersection of all cubes in the original benchmark `mark1` is nonempty; it gives a special case intersection pattern. We created a new benchmark called `mark1_11` from `mark1` by deleting five cubes in `mark1`. Notice that by deleting some cubes from the original benchmark, the new benchmark still has the property that the intersection of all its cubes is nonempty. The new test cases that we created are `mark1_11`, `mark1_12`, `mark1_13`, `mark1_14`, `mark1_15`, `shift_17`, `shift_18`, `shift_19`, and `shift_20`.

The experimental results on solving the special case λ-cube intersection problems are shown in Table 1. The second and the third column in the table list the number of cubes λ and the number of inputs n for each intersection problem, respectively. The fourth column lists the number of unknowns z_Γ's for each special case problem, which is equal to 2^λ. We solved the special case problem by applying Eq. (10). The fifth column of the table lists the runtime to solve each special case problem. Not surprisingly, the runtime increases exponentially with the number of cubes λ. This is because the number of unknowns z_Γ's increases exponentially with λ. However, the input to our program is an intersection pattern consisting of 2^λ numbers. Thus, in terms of the input size, the time complexity is linear. Further, for the benchmark `shift`, although the number of unknowns is more than 2 million, our algorithm was able to obtain the solution in about 70 s.

In the second set of experiments, we tested our algorithm that solves the general case problems. According to Algorithm 3, solving the general case problems involves two major steps. The first step is to check Statements 1 and 2 in Theorem 8 and establish Eq. (24). The second step is to solve Eq. (24) to obtain a non-negative integer solution. Since Eq. (24) is a set of linear equations and inequalities, it can be fed into a standard integer linear programming solver to obtain a non-negative integer solution or prove that such a solution does not exist. For this reason, we only focused on the first step. We developed a program that takes an intersection pattern, then checks Statements 1 and 2 in Theorem 8, and finally writes out Eq. (24).

[2] The intersection pattern benchmarks, which are shown in Tables 1 and 2, together with the sets of cubes that generate the intersection patterns, can be download from http://pan.baidu.com/s/1mgBn0yW.

Table 1
The experimental results on solving the special case λ-cube intersection problems.

Circuit	#cubes	#inputs	#unknowns	Time (s)
newtpla2	9	10	512	0
in3	10	35	1024	0
mark1_11	11	20	2048	0.01
mark1_12	12	20	4096	0.04
mark1_13	13	20	8192	0.08
mark1_14	14	20	16384	0.2
mark1_15	15	20	32768	0.48
mark1	16	20	65536	1.18
shift_17	17	19	131072	1.73
shift_18	18	19	262144	3.19
shift_19	19	19	524288	7.84
shift_20	20	19	1048576	24.97
shift	21	19	2097152	71.33

Table 2
The experimental results on solving the general case λ-cube intersection problems. The time reported is the time for checking Statements 1 and 2 in Theorem 8 and establishing Eq. (24). It does not include the time for solving Eq. (24).

Circuit	#cubes	#inputs	#unknowns					#equations			Time (s)
			Improved a	Basic b	Save (%) $(b-a)/b$	Naive c	Save (%) $(c-a)/c$	Improved d	Basic e	Save (%) $(e-d)/e$	
sqn	4	7	16	18	11.1	81	80.2	11	13	15.4	0
luc	6	8	66	74	10.8	729	90.9	32	40	20.0	0
br2	6	12	228	284	19.7	729	68.7	22	78	71.8	0
newcpla2	8	7	258	354	27.1	6561	96.1	65	161	59.6	0
newill	8	8	672	790	14.9	6561	89.8	39	157	75.2	0
tms	8	8	262	308	14.9	6561	96.0	69	115	40.0	0
prom2	9	9	512	767	33.2	19683	97.4	265	520	49.0	0.02
br1	10	12	8108	9113	11.0	59049	86.3	58	1063	94.5	0.12
vg2	10	25	1294	2248	42.4	59049	97.8	71	1025	93.1	0.01
exps	12	8	4130	4434	6.9	531441	99.2	399	703	43.2	0.09
alu1	12	12	4096	4100	0.098	531441	99.2	1300	1304	0.31	0.61
exp	14	8	69470	85162	18.4	4782969	98.5	122	15814	99.2	1.8
newtpla	14	15	127908	144197	11.3	4782969	97.3	117	16406	99.3	3.95
Average					17.1		92.1			58.5	

The experimental results on our program are shown in Table 2. The second and the third column in the table list the number of cubes λ and the number of inputs n for each intersection problem, respectively. We call our method as "improved", which generates Eq. (24). We listed the number of unknowns and the number of equations generated through the "improved" method for each benchmark. We compared the "improved" method with two other methods: the "basic" method, which establishes the system of equations (13)–(15), and the "naive" method, which takes all 3^{λ} combinations of column patterns as unknowns to set up equations. For each benchmark, we listed the number of unknowns needed by the "basic" method and the "naive" method and the number of equations needed by the "basic" method. We also listed the percentage of saving of the "improved" method on these metrics over the other two methods when proper. We can see that the "improved" method greatly reduces the number of unknowns and the number of equations. On average, it reduces the number of unknowns by 17.1% and the number of equations by 58.5% compared with the "basic" method. Compared with the "naive" method, it reduces the number of unknowns by 92.1%. The last column in the table listed the runtime for the "improved" method. Note that it is the time for checking Statements 1 and 2 in Theorem 8 and establishing Eq. (24). It does not include the time for solving Eq. (24).

7. Conclusion and future work

In this paper, we introduced a new problem, the λ-cube intersection problem: Given a set of numbers corresponding to an intersection pattern of a set of λ cubes, we are asked to synthesize a set of cubes to satisfy the given intersection pattern, or to show that there is no solution to the problem. We provide a rigorous mathematic treatment to this problem and derive a necessary and sufficient condition for the existence of a set of cubes to satisfy the given intersection pattern. The problem is reduced to checking whether a set of linear equations and inequalities has a non-negative integer solution.

As we mentioned in the introduction, a solution to the λ-cube intersection problem is an important step in solving the arithmetic two-level minimization problem. In our future work, we will apply the techniques proposed in this paper to develop a general solution to the arithmetic two-level minimization problem. For this purpose, we will look into another important subproblem, that is, to derive intersection patterns $(v_0, \ldots, v_{2^{\lambda}-1})$ on λ cubes that cover m minterms. The study in this work has offered us several important properties that we can use to derive proper intersection patterns. For example, the numbers v_Γ's must satisfy the Statements 1 and 2 in Theorem 8. Applying these properties will reduce the search space significantly. Finally, although the formulated problem can be theoretically solved by an integer linear programming (ILP) solver, the sizes of some large problems are beyond the capabilities of the state-of-the-art ILP solvers. However, we also notice that the formulated problem only asks whether a set of linear equations and inequalities has a non-negative integer solution. It does not have an objective function to optimize. Thus, it is not necessary to use an ILP solver to solve the formulated problem. In our future work, we will study the special structure of the set of linear equations and inequalities we derived in this paper and explore an efficient way to find a non-negative integer solution to it.

Acknowledgments

This work is supported in part by a National Science Foundation (NSF) CAREER Award, No. 0845650, NSF grant (No. CCF-1241987), and National Natural Science Foundation of China (NSFC) grant (No. 61204042). The authors would like to thank the anonymous reviewers for their constructive comments which greatly improved the quality of this work.

References

[1] Berkeley, 1993. Espresso. URL http://embedded.eecs.berkeley.edu/pubs/downloads/espresso/index.htm.
[2] C. Born, J. Kerbosch, Algorithm 457: Finding all cliques of an undirected graph, Commun. ACM 16 (9) (1973) 575–577.
[3] R.K. Brayton, G.D. Hachtel, C.T. McMullen, A.L. Sangiovanni-Vincentelli, Multilevel logic synthesis, Proc. IEEE 78 (2) (1990) 264–300.
[4] R.K. Brayton, C. McMullen, G.D. Hachtel, A. Sangiovanni-Vincentelli, Logic Minimization Algorithms for VLSI Synthesis, Kluwer Academic Publishers, 1984.
[5] W. Qian, M.D. Riedel, Two-level logic synthesis for probabilistic computation, in: International Workshop on Logic and Synthesis, 2010, pp. 95–102.
[6] W. Qian, M.D. Riedel, K. Barzagan, D. Lilja, The synthesis of combinational logic to generate probabilities, in: International Conference on Computer-Aided Design, 2009, pp. 367–374.
[7] R.L. Rudell, A. Sangiovanni-Vincentelli, Multiple-valued minimization for PLA optimization, IEEE Trans. Comput. Aided Des. 6 (5) (1987) 727–750.
[8] E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments, Theoret. Comput. Sci. 363 (1) (2006) 28–42.