UNIVERSIDAD DE CONCEPCIÓN

Facultad de Ciencias Físicas y Matemáticas

DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

k-Independencia en redes Booleanas

Tesis para obtener el título de Ingeniero Civil Matemático

Nombre:	Raúl Astete Elguin.
Carrera:	Ingeniería Civil Matemática.
Profesor Guía:	Dr. Julio Aracena Lucero.
Año:	2024

Agradecimientos

Agradezco en primer lugar a mi compañera de todo, Constanza, por su apoyo inquebrantable, dulzura y paciencia a lo largo de este proceso.

A mi madre-tía Carolina por su amor incondicional, su sacrificio y su constante apoyo. Todos mis logros siempre se los deberé en parte a ella, y ninguna secuencia de palabras será suficiente para poder expresar todo mi agradecimiento.

También quiero expresar mi agradecimiento a la familia de Constanza por abrirme las puertas de su hogar, su cálida acogida y generosidad han significado mucho para mí durante este tiempo.

A mi profesor guía Julio Aracena, por su sabiduría, responsabilidad y gentileza. Su orientación experta y su apoyo incondicional fueron fundamentales para alcanzar este logro académico.

A las profesoras y profesores del Departamento de Ingeniería Matemática, especialmente a Mónica Selva, Anahí Gajardo, Nicolás Sanhueza, Gabriel Gatica, Christopher Thraves, por su apoyo, enseñanzas y guía en esta carrera.

Quiero reconocer a mis amigos y compañeros de carrera, quienes, con su colaboración y confianza, han contribuido a crear un ambiente de aprendizaje en el que todos pudimos crecer juntos.

Al Centro de Investigación en Ingeniería Matemática (CI2MA) de la Universidad de Concepción, por permitirme utilizar sus espacios para trabajar y por brindarnos acceso a su cafetera. Su infraestructura, ambiente de trabajo y café fueron de gran ayuda para avanzar en este proyecto.

Agradezco a la Agencia Nacional de Investigación y Desarrollo (ANID) y al Centro de Modelamiento Matemático (CMM) por su apoyo financiero a través del proyecto Basal FB 210005.

Finalmente, agradezco a todas las personas que contribuyeron a hacer esta etapa de mi vida más llevadera.

Abstract

In this thesis, we define a new parameter for studying Boolean networks, called the "independence number". We establish that a Boolean network is k-independent if, for any set of k variables and any combination of binary values assigned to them, there exists at least one fixed point in the network that takes those values at the given set of k indices. In this context, we define the independence number of a network as the maximum value of k such that the network is k-independent.

This definition is closely related to widely studied combinatorial designs, such as "k-strength covering arrays", also known as Boolean sets with all k-projections surjective. Our motivation arises from understanding the relationship between a network's interaction graph and its fixed points, which deepens the classical paradigm of research in this direction by incorporating a particular structure on the set of fixed points, beyond merely observing their cardinality.

Specifically, we focus on studying interaction graphs that admit k-independent networks and show that the complete graph without loops with XOR-type activation functions achieves maximum nontrivial strength. Furthermore, we present constructions that demonstrate the existence of k-independent networks on n variables with disconnected, connected and strongly connected interaction graphs. We also study necessary conditions for a network to be k-independent.

Finally, we observe that computational simulations failed to find examples of monotone k-independent networks. This observation motivates Section 4.1 of this thesis, where we use another classical combinatorial design, called a Steiner system, to construct monotone k-independent networks with complete loopless interaction graphs.

Resumen

En esta tesis, definimos un nuevo parámetro para estudiar redes Booleanas, denominado “número de independencia”. Establecemos que una red Booleana es k-independiente si, para cualquier conjunto de k variables y cualquier combinación de valores binarios asignados a estas, existe al menos un punto fijo en la red que tome esos valores en dicho conjunto de k índices. En este contexto, definimos el número de independencia de una red como el máximo valor de k tal que la red es k-independiente.

Esta definición está estrechamente relacionada con diseños combinatorios ampliamente estudiados, como los “Covering arrays” de fuerza k, también conocidos como conjuntos Booleanos con todas las k-proyecciones sobreyectivas. Nuestra motivación surge de comprender la relación entre el grafo de interacción de una red y sus puntos fijos, lo que ayuda a profundizar en el paradigma clásico de las investigaciones en esta dirección al incorporar una estructura particular sobre el conjunto de puntos fijos, más allá de simplemente observar su cardinalidad.

Específicamente, nos centramos en estudiar grafos de interacción que admiten redes k-independientes, y mostramos que el grafo completo sin bucles con funciones de activación de tipo XOR alcanza la máxima fuerza no trivial. Además, presentamos construcciones que demuestran la existencia de redes k-independientes en n variables con grafos de interacción disconexos, conexos y fuertemente conexos.

Finalmente, observamos que las simulaciones computacionales no lograron encontrar ejemplos de redes monótonas k-independientes. Esta observación motiva la sección 4.1 de esta tesis, donde utilizamos otro diseño combinatorio clásico, llamado sistema de Steiner, para construir redes monótonas k-independientes con grafo de interacción completo y sin bucles.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Consider a group of n individuals who aim to reach a consensus on a decision, such as expressing their position as “In favor” or “Against”. These participants share their opinions in successive rounds, having access only to the opinions of their friends. Updates to their opinions for the next round are based on a predefined function. A fixed point is deemed to be achieved when the participant’s opinions stabilize, meaning that from that point onward, no further changes occur. For a fixed parameter $1 \leq k \leq n$, we analyze systems where any group of k individuals can make any of the 2^k possible configurations of choices, and for each of these decisions, there should exist a fixed point of the system with these states. We will term this property as k-independence.

This conceptualization will enable the modeling of systems whose dynamics exhibit a specific degree of local robustness in terms of achieving a stable state. This, in turn, opens up a broad array of pending questions regarding the characterization of these systems. In this work, we will focus on studying the characteristics of “decision” functions and the structure of the network that represents the interaction among entities in a k-independent system from a theoretical perspective.

In more specific terms, we will consider that the dynamics of these systems can be modeled with a Boolean Network, which is a system of n variables interacting with each other and evolving discretely over time according to a predefined rule. Boolean networks (BN) were originally introduced by Kauffman in 1969 [15] and have been widely used in various areas such as gene-regulatory networks, cryptography, and social systems [28], [30], [12]. Moreover, Boolean networks are an interesting combinatorial object in their own right, leading to numerous investigations into understanding their dynamic properties and their relationship with properties of their interaction graph. Furthermore, a discrete structure that will naturally emerge from our study are the so-called covering arrays (also studied as sets of vectors with all k-projections surjective, s-piercing sets of the hypercube and families of k-independent sets). The definition of k-independence presented in this work allowed us to connect these structures and incorporate a new perspective for the study of these topics, contributing to the state-of-the-art by laying the foundation for a new research subject.

1.2 Related work and contribution

As we will see in the following chapters, the notion of k-independence that we have defined is equivalent to establishing that the set of fixed points of a Boolean network forms a covering array of strength k. In other words, this set consists of vectors in $\{0,1\}^n$ such that for any subset of k variables and any assignment of zeros and ones, there is at least one fixed point of the network that assumes that configuration in that subset.

It is of interest to understand, at a theoretical level, the configurations that lead a Boolean network to stabilize, that is, periodic points [29, 10], meaning the states $x \in \{0,1\}^n$ such that $f^{\ell}(x) = x$ for some ℓ. Fixed points (case $\ell = 1$) are particularly interesting for inferring information about the activation functions of the network [19]. However, most works in this direction study the relationship between the number of fixed points of a BN and the properties of the local activation functions [2, 4] or of its interaction graph. The information that can be obtained about the architecture of a Boolean network from **structural** properties of its fixed points has not been thoroughly explored. A first step in this direction is the work carried out in [24], where the VC dimension in Boolean networks is defined in terms of their fixed points.

As we mentioned previously, this work primarily focuses on the concepts of covering arrays and Boolean networks. Our aim is to dive deeper into the fixed points of a Boolean network, not only examining their quantity, but also endowing this set with a specific structure. An interesting contrast lies in the fact that the central question in the study of covering arrays is predominantly quantitative. It involves determining, given n and k, the covering array with the minimum number of rows that achieves strength k.

Our contribution begins by identifying the necessary conditions to achieve k-independence in terms of local activation functions, the number of fixed points, and properties of the interaction graph. Subsequently, we work on demonstrating the existence of k-independent networks with graphs belonging to certain classical families, showing one disconnected family, another connected, and another strongly connected. Additionally, we identify specific cases where we know sufficient conditions, in terms of the interaction graph, for k-independence. Furthermore, computational work was conducted to develop basic algorithms for generating examples of k-independent networks. Within this endeavor, a thorough review of all graphs up to 7 vertices capable of being interaction graphs of k-independent networks, with $k \geq 2$ and fixed activation functions, was carried out. This allowed for the creation of a catalog of small interesting graphs and the observation that clarity was lacking regarding the existence of monotone k-independent networks with $k \geq 2$. Consequently, our work also includes the construction of a family of k-independent networks, whose set of fixed points originates from a certain type of combinatorial design known as a Steiner system.

Chapter 2

Definitions and basic concepts

In this section, firstly, we will review basic definitions of some concepts in discrete mathematics. Subsequently, we will delve into two central concepts of this work: Covering arrays and Boolean networks. Finally, this chapter will include some elementary propositions and lemmas that will be used in the following chapters.

2.1 Discrete mathematics concepts

Given an integer $n \geq 1$ we denote by $\{0,1\}^n$ the set of n-length vectors consisting of ones or zeros. We denote $\vec{0}$ as the vector with all components being zero and $\vec{1}$ as the vector with all components being one. In addition, we establish the notation $[n] := \{1,\ldots,n\}$ and given $x \in \{0,1\}^n$ and $J = \{j_1,\ldots,j_t\} \subseteq [n]$, we define the projection of x into J as the vector $x_J = (x_{j_1},\ldots,x_{j_t}) \in \{0,1\}^{|J|}$. Given $x \in \{0,1\}^n$ we define the Hamming weight $w_H(x)$ (or simply weight) as the number of ones it contains.

Definition 1. *A function $f : \{0,1\}^n \to \{0,1\}$ is said to be a Boolean function on n variables. It is worth mentioning that every Boolean function can be described as a logical expression in the variables $x_1,\ldots,x_n$, utilizing conjunction (represented by the symbol $\wedge$), disjunction (represented by the symbol $\vee$), and negation (represented by a bar above the variable, e.g., $\overline{x}$). Moreover, we define:*

- *A literal is a variable or its negation.*
- *A clause is a set of literals connected by the symbol $\vee$ or $\wedge$.*
- *A Conjunctive Normal Form (CNF) of f has disjunctive clauses connected by $\wedge$, and a Disjunctive Normal Form (DNF) has conjunctive clauses connected by $\vee$.*

Definition 2. *A graph G is a pair (V,E), where V is a non-empty finite set and E is a subset of $\{e \in 2^V : |e| = 2\}$. The elements of V are usually called vertices or nodes, and the elements of E edges. We say that a sequence of distinct vertices $v_1,\ldots,v_k$ is a path if for every consecutive pair* *of vertices v_iv_{i+1} is an edge of G. A graph G is connected if for every pair of vertices v_i, v_j, there is a path connecting them.*

Definition 3. *A directed graph (or digraph) D consists of a pair (V, A), where $V = \{v_1, \dots, v_n\}$ is a non-empty finite set of nodes or vertices, and a set $A \subseteq V \times V$ containing ordered pairs of vertices, known as arcs. For a given vertex $v \in V$, we introduce the concepts of in-neighborhood $N^-(v)$ and out-neighborhood $N^+(v)$. These are defined as follows:*

$$N^-(v) = \{u \in V \,:\, (u, v) \in A\}, \quad N^+(v) = \{u \in V \,:\, (v, u) \in A\}.$$

Furthermore, we define the out-degree of the vertex v as $d^+(v) = |N^+(v)|$ and the in-degree as $d^-(v) = |N^-(v)|$. Also, we define the minimum in-degree as $\delta^-(D) = \min\{d^-(v) \,:\, v \in V\}$ and similarly $\delta^+(D) = \min\{d^+(v) \,:\, v \in V\}$.

Definition 4. *Given a graph $G = (V, E)$ and $v \in V$, we define the neighborhood of v as $N(v) = \{u \in V \,:\, uv \in E\}$. We also define $d(v) = |N(v)|$, $\delta(G) = \min\{d(v) \,:\, v \in V\}$, and we say that G is k-regular if every vertex has degree k.*

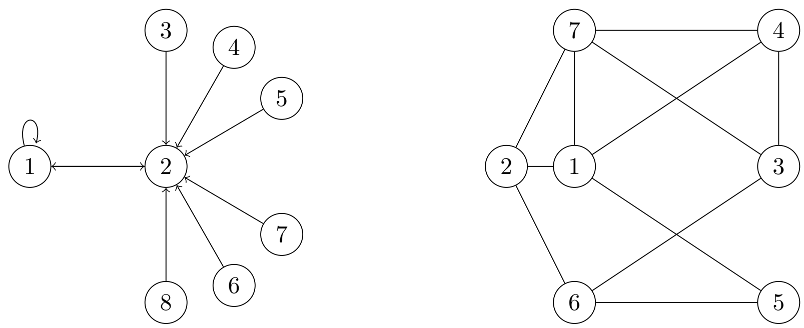


Figure 2.1. Examples of a digraph (left) and graph (right).

Remark 1. *As we can see in the left digraph in Figure 2.1 there can also be arcs from a vertex to itself. We will refer to such arcs as loops. On the other hand, it is also possible to have an arc (i, j) and simultaneously an arc (j, i). In such a case, we say that the interaction between i and j is symmetric and we may draw the line without an arrowhead. In such a case, we will also refer to the arcs as "edges". In this work, when we refer to graphs, we will be talking about digraphs in which all interactions are symmetric.*

Definition 5. *A path in a digraph $D = (V, A)$ is a sequence of distinct vertices $v_1, \dots, v_k$ such that for every $i = 1, \dots, k-1$, (v_i, v_{i+1}) is an arc of D. Additionally, we define the fundamental graph of D, denoted as $D_F = (V, E)$, where $\{u, v\} \in E$ if $(u, v) \in A$ or $(v, u) \in A$. Subsequently, we state that D is connected if D_F is connected, and we declare D to be strongly connected if, for every pair of vertices $v_i, v_j \in V$, there exists a path P from v_i to v_j and a path $\tilde{P}$ from v_j to v_i.*

We define the complete graph (without loops) on n vertices as a graph $K_n = (V, A)$ such that $|V| = n$ and $A = V \times V \setminus \{(i, i) \,:\, i \in V\}$. In other words, the complete graph is one that includes all possible edges, for example, see Figure 3.2. Also, given an integer n, we will denote by Q_n the graph that can be described by the vertices $V(Q_n) = \{0, 1\}^n$, where two vertices are adjacent if they differ in exactly one component. This graph is known as the n-cube or n-dimensional hypercube.

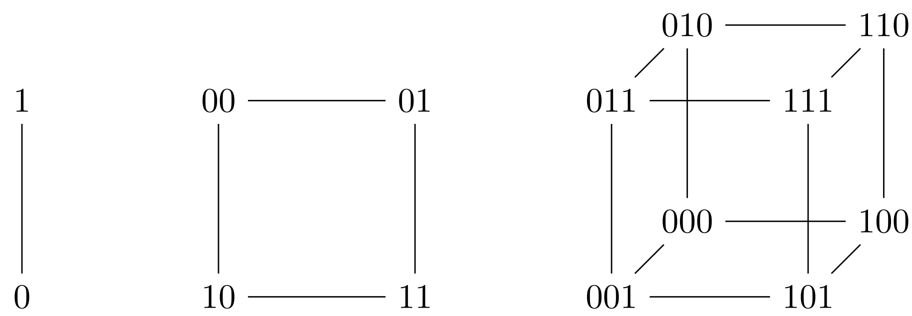


Figure 2.2. Q_1, Q_2 and Q_3

2.2 Covering arrays

In this section, we will introduce a combinatorial concept that naturally emerges from the concept of k-independence that we want to study. Let us suppose that we have an algorithm $A(x_1, \ldots, x_n)$ with n binary inputs and that we know that this algorithm fails due to the interaction of k unknown variables. So, our objective is to test the algorithm to identify these faulty parameters. A naïve approach would be to test every possible combination of these k parameters, resulting in $\binom{n}{k}2^k$ tests. Unfortunately, this number can be too large for reasonably small values of n and k, making this solution impractical. To address this, a covering array of strength k is defined as a set of Boolean vectors from $\{0, 1\}^n$ such that for every subset I of k indices, and for every $a = (a_1, \ldots, a_k) \in \{0, 1\}^k$, there exists a vector x in the set such that $x_I = a$. Also, we denote $CA(m, n; k)$ as the set of all covering arrays with m vectors of size n and strength k. When we do not need to refer to the number of rows, we simply denote it by $CA(n; k)$. For example, the following is an element of $CA(5, 4; 2)$:

$$B = \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1. \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array}$$

Since we want to perform as few tests as possible, we are looking for covering arrays with the minimum number of rows. So we define by $CAN(n; k)$ the minimum number of rows of a matrix in $CA(m, n; k)$. It is worth mentioning that determining $CAN(n; k)$ for arbitrary values of n and k remains an open problem; we can see some of the known values in Table 2.1. Various efforts have been made to find approximations to this minimum. However, the case of $k = 2$ is the only one that has been completely solved [18].

Theorem 1 (Minimal construction for strength 2, [18])**.** *Let n be an integer. Then $CAN(n; 2)$ is* *the minimum number m such that*

$$\binom{m-1}{\lceil \frac{m}{2} \rceil} \geq n$$

and it is possible to construct a covering array with these parameters by adding as columns all possible vectors from $\{0,1\}^{m-1}$ of weight $\lceil m/2 \rceil$ and adding the zero vector.

In a more general case, from [18] and [26] we know the bounds:

$$\Omega(2^k \log n) \leq CAN(n;k) \leq \frac{k}{\log\left(\frac{2^k}{2^k-1}\right)} \log n. \tag{2.1}$$

In other words, for a fixed k, $CAN(n;k) = \Theta(\log n)$, justifying the use of covering arrays as testing objects and the search for those with the least possible number of rows. For this reason, various methods for their construction have been studied. Some of them include incremental construction methods [27], using error-correcting codes [1], derandomized algorithms [7, 6], and other heuristic-based methods [23].

$s \backslash t$	1	2	3	4	5	6
0	2	4	8	16	32	64
1	2	4	8	12	32	64
2	2	5	10	21	42	85
3	2	6	12	24	48-52	96-108
4	2	6	12	24	48-54	96-116
5	2	6	12	24	48-56	96-118
6	2	6	12	24	48-64	96-128
7	2	6	12	24	48-64	96-128
8	2	6	12	24	48-64	96-128
9	2	7	15	30-32	60-64	120-128
10	2	7	15-16	30-35	60-79	120-179

Table 2.1. Some known values of $CAN(s+t;t)$ [20].

For a fixed k, as we increase n, CAN also increases, as seen in Table 2.1. We can understand that this happens because given an element in $CA(n+s;k)$, we can eliminate s columns and obtain an element in $CA(n;k)$. On the other hand, it is important to note that the lower bound of (2.1) is a necessary but not sufficient condition. To see this consider any set of vectors with more than $\Omega(2^k \log n)$ elements but with the i-th component equal to zero in every vector. In such a case, this array would have strength zero.

Regarding the complexity of combinatorial problems associated with covering arrays, there are still mostly open problems, and the state of the art is somewhat immature as there have been several instances of incorrectly formulated or misinterpreted propositions concerning the NP-hardness of generating such sets. However, some closely related problems, being NP-complete suggest that finding a minimal covering array is a tough optimization challenge. In [14], a review of the state of the art in the complexity of these problems is presented.

2.3 Some different formulations

Recall that the n-dimensional hypercube, often referred to as the n-cube (or cube, when n is clear), is a graph where the vertices are labeled with vectors from $\{0,1\}^n$, and there is an edge connecting every pair of vertices that differ in exactly one component. This concept allows us to characterize covering arrays in terms of the structure of the cube.

Definition 6. *Given an hypercube graph Q_n, $S \subseteq V(Q_n)$ is a s-piercing set if $Q_n - S$ does not contain Q_s as a subgraph. Then we have*

$$S \subseteq V(Q_n) \text{ is a } s\text{-piercing set} \iff S \in CA(|S|, n; n-s)$$

Example 1. *Consider $n = 3$, $s = 1$, and let $S = \{000, 110, 101, 011\}$ be a covering array of strength 2. In this particular example, it is observed that $Q_3 - S$ has no edges, corresponding to Q_1 copies. Therefore, it is a 1-piercing set.*

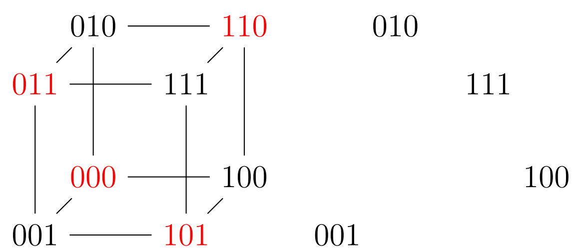


Figure 2.3. Q_3 and $Q_3 - S$, with $S \in CA(4, 3; 2)$

The previous formulation facilitates deriving clearer upper bounds for $CAN(n; k)$, along with describing elementary cases where $CAN(n; k)$ is known exactly. An illustrative example of this is provided by the following proposition from [11].

Proposition 1. *[11] For $n > k > 1$, $j \in \{0, 1, \ldots, n-k\}$, we have*

$$S_j = \{x \in \{0,1\}^n \; : \; w_H(x) = j \mod (n-k+1)\} \in CA(n; n-k).$$

Proof. Note that we can partition the vertices of any Q_k subcube with the $k+1$ sets $\{x \in \{0,1\}^n : w_H(x_{Q_k}) = i\}$, $0 \leq i \leq k$. Each of these sets will be called a level. Then we want to prove that the removal of S_j leaves Q_n without copies of Q_k. Hence, by eliminating one level from any of the $k+1$ consecutive ones, no occurrence of Q_k remains. Fixing the weight modulo $n-k+1$ produces this effect. $\square$

Definition 7. *A family $\mathcal{F} = \{F_1, \ldots, F_n\}$ of subsets of a set is said to be k-independent if for every pair of disjoint subsets S_1, S_2 of $[n]$ such that $|S_1| + |S_2| = k$, we have that*

$$\left(\bigcap_{i \in S_1} F_i\right) \cap \left(\bigcap_{j \in S_2} \overline{F_j}\right) \neq \emptyset$$

Example 2. *The family $\mathcal{F} = \{\{4,6,7,8\},\{3,5,7,8\},\{2,5,6,8\},\{1,5,6,7\}\}$ is 3-independent. As we will see in the following proposition, one way to check that this family is 3-independent, is to check that rows of the following matrix forms a covering array of strength 3:*

	1	2	3	4
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
5	0	1	1	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

Proposition 2. *Let $A = \{x^1, \ldots, x^m\}$ be a set of m Boolean vectors from $\{0,1\}^n$, and denote by $\mathcal{F}_A = \{F_1, \ldots, F_n\}$ the family of subsets from $[m]$ given by $F_i = \{j \in \{1,\ldots,m\} \,:\, x_j^i = 1\}$. Then, $A \in CA(m,n;k)$ if and only if $\mathcal{F}_A$ is k-independent.*

Proof. First let A be a covering array of strength k, and consider the family $\mathcal{F}$ already defined. Let S_1, S_2 disjoint subsets of $[n]$ such that $|S_1| + |S_2| = k$. Let us suppose without loss of generality that $|S_1| = \ell$, and denote $S_1 = \{1,\ldots,\ell\}$ and $S_2 = \{\ell+1,\ldots,k\}$. Also, denote $I = \{1,\ldots,k\}$. Now we consider the configuration $a = (a_1,\ldots,a_k) \in \{0,1\}^k$ such that $a_1 = \cdots = a_\ell = 1$ and all the components between positions $\ell+1$ and k are zero. Then, since A has strength k, there exists $x \in A$ such that $x_I = a$. This implies $\left(\bigcap_{i\in S_1} F_i\right) \cap \left(\bigcap_{j\in S_2} \overline{F_j}\right)$ is nonempty and therefore, $\mathcal{F}_A$ is k-independent.

Conversely, suppose $\mathcal{F}_A$ be a k-independent family of subsets, and define $A = \{x^1,\ldots,x^m\}$ such that $x_i^\ell = 1$ if and only if $\ell \in F_i$. Now let $I = \{i_1,\ldots,i_k\} \subseteq [n]$ and $a = (a_1,\ldots,a_k) \in \{0,1\}^k$, and define $S_1 = \{i_j \in I \,:\, a_j = 1\}$, $S_2 = \{i_j \in I \,:\, a_j = 0\}$. Then, $|S_1| + |S_j| = k$, and since $\mathcal{F}_A$ is k-independent, there exists $j \in \{1,\ldots,m\}$ such that for every $i \in S_1$, $x_i^j = 1$ and for every $i \in S_2$, we have $x_i^j = 0$. In other words, x^j takes the values given by a in the components indexed by I. Since both of them are arbitrary, we conclude A is a covering array of strength k. $\square$

Remark 2. *If we have a covering array $A = \{x^1,\ldots,x^m\}$ and we represent it through the following array:*

$$A = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^m \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^m & x_2^m & \cdots & x_n^m \end{bmatrix}$$

Then, it is clear from Proposition 2 that we can perform permutations by rows and columns, as well as switches (changing ones to zeros and zeros to ones) by columns, and still have a covering array with the same parameters. Thus, two covering arrays A and B are said to be isomorphic if one can be obtained from the other using these three operations. Since we are working with covering arrays as sets, we do not care about the order of the rows. Finally, it is worth noting that the size of the equivalence class of a covering array is $n!2^n$, considering the $n!$ column permutations and 2^n column switches.

2.4 Boolean networks

On the other hand, as mentioned earlier, a Boolean network is a system of n variables that interact with each other and evolve discretely over time according to a predefined rule. Boolean Networks (BNs or simply networks) were originally introduced by Kauffman in 1969 [15]. In a formal sense, a Boolean network can be expressed as a function $f : \{0,1\}^n \to \{0,1\}^n$, where $f(x) = (f_1(x), \ldots, f_n(x))$ for $x \in \{0,1\}^n$. Here, each Boolean function of the type $f_i : \{0,1\}^n \to \{0,1\}$ is referred to as a local activation function of the network. Boolean networks are a particular case of a discrete dynamical system, and we can understand their dynamics as described by successive iterations. In this context, the iteration digraph of a Boolean network f over the vertices $\{0,1\}^n$ is defined such that the arcs are of the form $(x, f(x))$ for $x \in \{0,1\}^n$. Each iteration digraph fully represents a BN; however, their utilization becomes impractical due to their large number of nodes. For this reason, associated with any Boolean network f, we can define the interaction (or dependency) digraph $G(f)$, with vertices $\{1, \ldots, n\}$ and arcs (i, j) indicating that f_j "depends" on variable i, i.e., there exists $x \in \{0,1\}^n$ such that

$$f_j(x_1, \ldots, x_i = 0, \ldots, x_n) \neq f_j(x_1, \ldots, x_i = 1, \ldots, x_n).$$

It is important to note that $G(f)$ may have loops, i.e., arcs from a vertex to itself. A fixed point of f is a vector $x \in \{0,1\}^n$ such that $f(x) = x$. We will denote the **set of fixed points** by $FP(f) = \{x \in \{0,1\}^n \,:\, f(x) = x\}$. The set of fixed points in a BN is an intriguing subject of study for various reasons. One of them is its significance in applications within biological systems, as they can be interpreted as stable patterns of gene expression [16]. However, as previously mentioned, there are very few studies that address the set of fixed points from a qualitative perspective. Therefore, we propose the following definition:

Definition 8. *Given two integers $1 \leq k \leq n$, we say that a Boolean network $f : \{0,1\}^n \to \{0,1\}^n$ is k-independent if for every $I \subseteq [n]$ of size k and for all $a = (a_1, \ldots, a_k) \in \{0,1\}^k$, there is some fixed point $x \in \{0,1\}^n$ such that $x_I = a$. It is easy to see that f is k-independent if and only if $FP(f) \in CA(m, n; k)$ with $m = |FP(f)|$. We define the independence number of f, denoted as $i(f)$, as the maximum k for which f is k-independent.*

Example 3. *The Boolean network $f : \{0,1\}^3 \to \{0,1\}^3$ defined by*

$$f_1(x) = x_1 \wedge (x_2 \vee x_3)$$
$$f_2(x) = x_2 \wedge (x_1 \vee x_3)$$
$$f_3(x) = x_3 \wedge (x_1 \vee x_2)$$

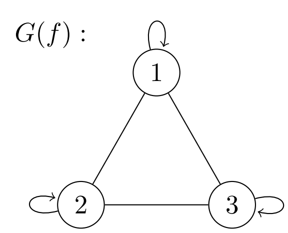


Figure 2.4. Example of a BN with $i(f) = 2$.

has fixed points FP$(f) = \{000, 010, 110, 101, 111\}$ which forms a covering array of strength 2. So f is 2-independent. Moreover, since f is not the identity map, it cannot be 3-independent. Therefore, $i(f) = 2$.

In order to explore the family of digraphs that allow some k-independent Boolean networks to be compatible with it as its dependency graph, we propose the following definition:

Definition 9. *Let $k \leq n$ be an integer and G a (possibly) directed graph on n vertices. We say that G is k-admissible if there exists a k-independent Boolean network f such that $G(f) = G$. For example, the graph from Figure 2.4 is 2-admissible.*

Chapter 3

About the interaction graph of *k*-independent Boolean networks

In this section, we will establish the basic results on the k-admissibility of graphs and the k-independence of arbitrary Boolean networks. To do this, first, we will review some classical results from the literature concerning fixed points of Boolean networks. As we have already mentioned, a significant motivation in this area is to answer the question: What can we infer about the fixed points of f based on $G(f)$, and vice versa? The results we will present initially compare the number of fixed points of f with properties of $G(f)$. Perhaps the most referenced result in this field is the feedback bound.

Let us recall that, given a directed graph $G = (V, A)$, we define a set $S \subseteq V$ as a feedback vertex set if the subgraph $G[V \setminus S]$ is acyclic. Furthermore, we introduce the transversal number of G, denoted by $\tau(G)$, as the minimum cardinality of a feedback vertex set for G.

Theorem 2 (Feedback bound; Aracena 2008, [2])**.** *For any Boolean Network f we have:*

$$|FP(f)| \leq 2^{\tau(G(f))}$$

This result establishes a necessary condition for the k-admissibility of graphs. Specifically, for a graph G to be k-admissible, it must be the interaction graph of a Boolean network, where the fixed points form a covering array of strength k. This necessitates having at least 2^k fixed points. Furthermore, we require that

$$CAN(n;k) \leq 2^{\tau(G)} \iff \tau(G) \geq \log CAN(n;k)$$

It is important to note that for some values of n and k, as seen in Table 2.1, $\log CAN(n;k) > k$, and therefore in such situations, k-admissible graphs require $k < \tau(G)$.

Example 4. *Consider a complete bipartite graph $K_{n,2}$. In this case, $\tau(K_{n,2}) = 2$. Then, the feedback bound allows us to establish that for any Boolean network f with interaction graph $K_{n,2}$, $|FP(f)| \leq 2^2 = 4$. Later, as we have already seen in Table 2.1, for all $n \geq 4$ we have $CAN(n;2) > 4$, we can conclude that for $n \geq 4$, $K_{n,2}$* ***is not*** *k-admissible for any $1 < k \leq n$.*

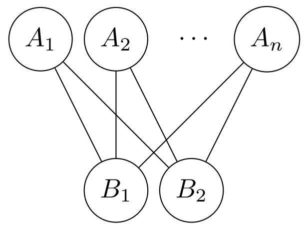


Figure 3.1. Complete bipartite graph $K_{n,2}$ with a partition of size 2.

It might be intuitive to think that for a graph G to be k-admissible, it is necessary that $\tau(G) > k$. However, there are examples where for all $n \geq 3$ and $k = 1, \ldots, n-1$, G is k-admissible with $k \leq \tau(G) = n - 1$. This is the case when G is a loopless complete graph.

Lemma 1. *Let $G = K_n$ be the complete graph without loops. Then G is $(n-1)$-admissible.*

Proof. Let $f : \{0,1\}^n \to \{0,1\}^n$. Now we want to characterize the set of fixed points of f. Let $x \in \{0,1\}^n$ be a fixed point of f. Now let $i \in [n]$. If $x_i = 0$, we need that i observes an even number of ones, so $w_H(x)$ should be even. Otherwise, if $x_i = 1$, the component i observes an odd number of ones, so $w_H(x)$ should also be even. We conclude that

$$FP(f) = \{x \in \{0,1\}^n : w_H(x) = 0 \mod 2\}.$$

Now, by using Proposition 1 with $j = 0$ and $k = n-1$, $FP(f) = S_0$ is a covering array of strength $t = n - 1$. $\square$

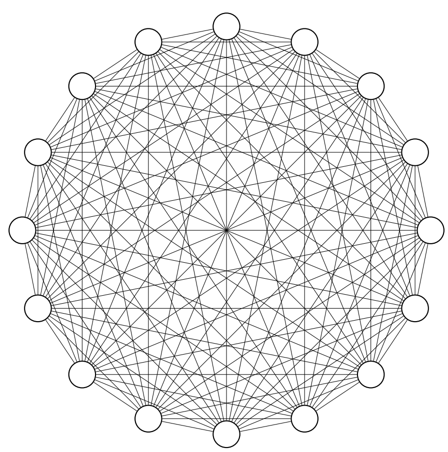


Figure 3.2. K_{16} is 15-admissible.

In order to find examples of k-admissible graphs for different values of k, we used the following procedure:

1) Generate a random matrix A (i.e., selecting every entry to be 0 or 1 with probability 1/2) until it is a covering array of strength k. The number of rows is manually set based on the computation time of the generation since $CAN(n; k)$ is not known in the general case.

2) Randomly, fill the table of a Boolean Network that only has A as the set of fixed points. If another fixed point is generated, then repeat the random assignment.

3) Return the Boolean Network.

Using this procedure, we observe that for $k \geq 2$ the resulting interaction graph is (almost) always the complete graph. This led us to believe that k-independence required a complete interaction graph. However, we discovered counterexamples that refute this statement, as illustrated in Example 5.

Through extensive brute-force computational simulations, we found that identifying k-independent networks poses a considerable challenge. As a result, we conjecture that the proportion of networks exhibiting k-independence is very small. Supporting this conjecture is the fact that the expected number of fixed points in a Boolean network, whose Boolean functions are drawn from probability distributions not necessarily uniform or identical, is one. In contrast, we know that k-independent networks require at least 2^k fixed points.

It is worth noting that the conjecture that k-independence for $k \geq 2$ implied a complete interaction graph (including loops) stemmed from the bias of randomly and independently completing the entries of the boolean network's table. Indeed, let's assume that f is a BN whose table was chosen for each entry with a probability of $1/2$. Let A_{ij} denote the event "f_j does not depends on x_i", i.e., the arc (i, j) does not exists in $G(f)$. Then, we can establish the following estimate:

$$\begin{aligned}
\Pr(A_{ij}) &= \Pr\left(\forall x \in \{0,1\}^n \,:\, f_j(x \,:\, x_i = 0) = f_j(x \,:\, x_i = 1)\right) \\
&= \prod_{x \in \{0,1\}^n} \Pr(f_j(x \,:\, x_i = 0) = f_j(x \,:\, x_i = 1)) \\
&= \prod_{x \in \{0,1\}^n} \frac{1}{2} \\
&= \frac{1}{2^{2^n}}
\end{aligned}$$

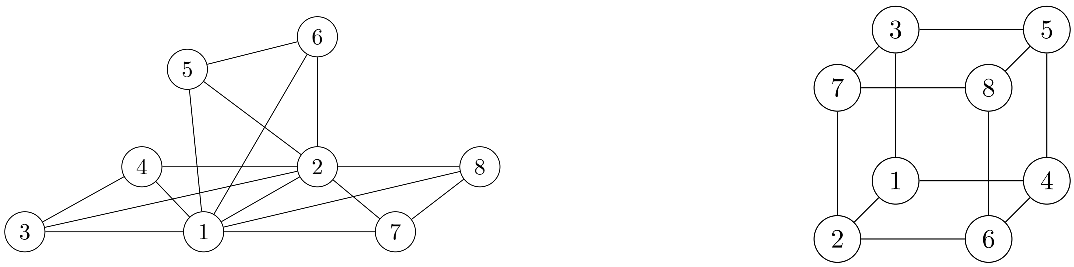
Therefore,

$$\Pr(\overline{A_{ij}}) = \Pr[(i,j) \in A(G(f))] = 1 - \Pr(A_{ij}) = 1 - \frac{1}{2^{2^n}}$$

In consequence,

$$\Pr(G(f) \text{ is complete}) = \prod_{i=1}^{n} \prod_{j=1}^{n} \Pr(\overline{A_{ij}}) = \left(1 - \frac{1}{2^{2^n}}\right)^{n^2} \xrightarrow{n \to \infty} 1.$$

Example 5. *The following are examples of* 3*-admissible graphs that are not the complete graph.*



a) A 3-admissible graph with XOR

b) 3-cube graph, 3-admissible with XOR

Figure 3.3. Two examples of 3-independent BNs with interaction graph non complete

The example (a) was discovered by combining 3 copies of K_4 in a way that preserved fixed points with a force of 3. Conversely, example (b) emerged during the exploration of classical graphs while imposing the XOR function. There are two primary reasons for concentrating on scenarios where all local activation functions are of the XOR type. The first reason is that we already know cliques with XOR functions generate the maximum possible force without trivializing the network. Furthermore, networks of this kind typically exhibit a higher number of fixed points compared to others, such as AND-OR networks [3].

While we have already seen that k-admissible graphs, with $k \geq 2$, are not necessarily complete, it is true that they tend to become denser for larger values of k. In fact, to prove this, let us first consider the following definition.

Definition 10 (k-set canalizing functions [13])**.** *We say that $h : \{0,1\}^n \to \{0,1\}$ is k-set canalizing if there exists a set $I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ and values $a_1, \dots, a_k, b \in \{0,1\}$ such that*

$$\forall x \in \{0,1\}^n,\ x_I = (a_1, \dots, a_k) \implies h(x) = b$$

In this context, we say that the input $a_1, \dots, a_k$ canalizes h to b. Moreover, we denote by $IC(h)$ the minimum k such that h is k-set canalizing.

Proposition 3. *Let h be a Boolean function. Then, $IC(h) = k$ if and only if the minimum number of literals in a clause of a dnf-formula of h is k.*

Proof. Let h be a Boolean function such that we can write one of its dnf-formula as

$$h(x_1, \dots, x_n) = \bigvee_{i=1}^{\ell} C_i$$

where each clause $C_i = (y_{i_1} \wedge y_{i_2} \wedge \cdots \wedge y_{i_{k(i)}})$ involves $k(i)$ literals and every y_{i_ℓ} is x_{i_ℓ} or its negation. Also, suppose $k = \min\{k(i) : i = 1, \dots, \ell\}$. Therefore, there exists a clause $C_i = (y_{i_1} \wedge y_{i_2} \wedge \cdots \wedge y_{i_k})$. Now define $a = (a_1, \dots, a_k) \in \{0,1\}^k$ by $a_j = 1$ if $y_{i_j} = x_{i_j}$ and $a_j = 0$ if $y_{i_j} = \overline{x_{i_j}}$. Then, h is k-set canalizing and canalizes to 1 in the input a.

Conversely, suppose h is k-set canalizing and w.l.o.g. that $I = \{1, \dots, k\} \subseteq [n]$ is the set of indexes of the canalizing variables. Then, if we construct the dnf-formula of f from its table, we can write

$$h(x_1, \dots, x_n) = g(x_1, \dots, x_n) \vee (x_1 \wedge \cdots \wedge x_k)$$

So, $IC(h) \leq k$. By contradiction if we suppose there is a smaller clause, with $\ell < k$ literals, h would be ℓ-set canalizing, which is a contradiction. Finally, we conclude $IC(h) = k$. $\square$

Example 6. *The following are examples of k-set canalizing functions:*

- *The function $g : \{0,1\}^n \to \{0,1\}$, defined as $f(x_1, \dots, x_n) = \bigwedge_{i=1}^n x_i$, is 1-set canalizing. It canalizes to zero whenever any variable takes the value zero. Similarly, disjunctions are 1-set canalizing, channeling to one when any variable takes the value one.*
- *The function $b_n^k : \{0,1\}^n \to \{0,1\}$, defined as $b_n^k(x_1, \dots, x_n) = 1$ if and only if $w_H(x) = k$, is known as a **belt function** with $m = 1$[22]. There are two cases: if $k \leq \lfloor n/2 \rfloor$, then $IC(b_n^k) = k$, as it canalizes to 1 when encountering k ones. Otherwise, if $k \geq \lfloor n/2 \rfloor$, $IC(b_n^k) = n - k + 1$, as it canalizes to zero when encountering $n - k + 1$ zeros. If $n - k + 1$ components are zero, at most $k - 1$ components are one, making it impossible for the output to be 1.*

With this setting, we are now ready to establish the following result:

Theorem 3. *Let $f = (f_1, \dots, f_n)$ be a k-independent Boolean networks such that $G(f)$ has no loops, then for all i, $IC(f_i) \geq k$.*

Proof. By contradiction, let's assume that f is k-independent, and that there exists a local activation function f_i that canalizes into $\tilde{I} = \{i_1, \dots, i_\ell\} \subseteq N^-(i)$ with $\ell < k$, on inputs $a = (a_1, \dots, a_\ell) \in \{0,1\}^\ell$ to the value $b \in \{0,1\}$. Since there are no loops, we may assume that $i \notin \tilde{I}$. Then, $|\tilde{I} \cup \{i\}| = \ell + 1 \leq k$, and since f is k-independent (and thus $(\ell+1)$-independent), there exist two fixed points $x, y \in FP(f)$ such that:

$$x_i = 0, y_i = 1, x_{\tilde{I}} = a = y_{\tilde{I}}$$

Therefore, $f_i(x) = f_i(y) = b$, but $f_i(x) = x_i = 0$ and $f_i(y) = y_i = 1$, which is a contradiction. $\square$

Corollary 3.1. *If G is a loopless k-admissible digraph, then its minimum indegree is at least k.*

Corollary 3.2. *There is no AND-OR Boolean network f with $i(f) \geq 2$ if $G(f)$ has no loops.*

The preceding result underscores that there exist networks with an exponential number of fixed points, all the while maintaining $i(f) = 1$. In [3], we encounter the following compelling example.

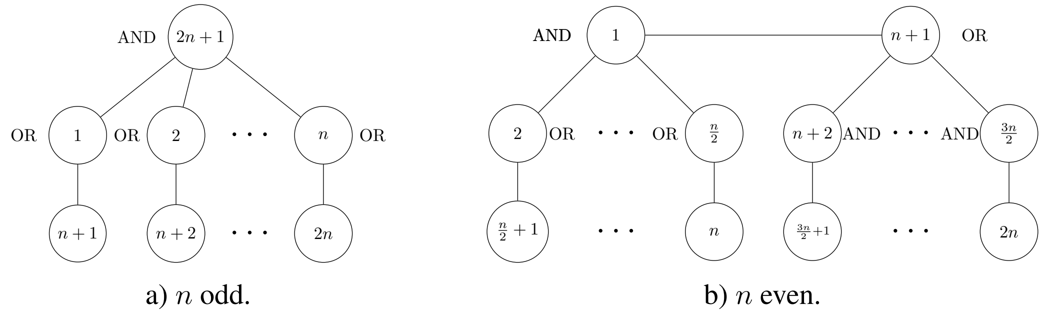


Figure 3.4. AND–OR networks realizing the maximum number of fixed points.

We denote by m the number of nodes of the network. The above example achieves $2^{(m-1)/2}$ fixed points for m odd and $2^{(m-2)/2}+1$ for the even case. This is the maximum number of fixed points that an AND-OR Boolean network can have and for both cases $\tau(G(f)) = \lfloor \frac{m}{2} \rfloor$. However, since disjunctions and conjunctions are 1-set canalizing, the set of fixed points cannot have strength greater than 1.

Remark 3. *It is worth mentioning that the hypothesis of having no loops is necessary to conclude the previous results. For instance, let's consider network $f : \{0,1\}^{n+1} \to \{0,1\}^{n+1}$ defined by $f_i(x) = x_i$, for $i = 1, \dots, n$; and*

$$f_{n+1}(x) = x_{n+1} \vee \left(\bigwedge_{i=1}^{n} \overline{x_i} \right),$$

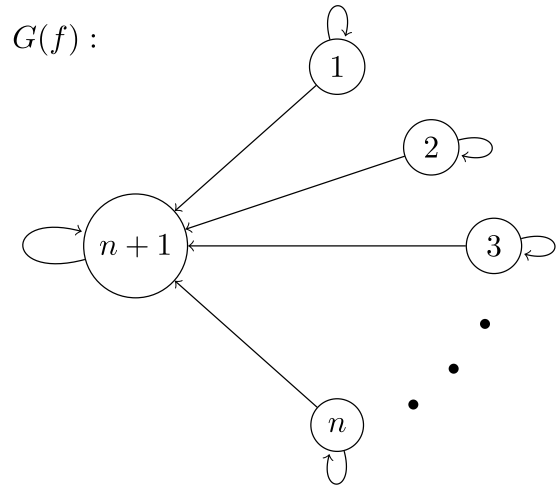


Figure 3.5. Example where loops are present, and the previous results are not valid.

Then, it is easy to see that the set of fixed points of f is $\{0,1\}^{n+1} \setminus \{\vec{0}\}$, and this set is a covering array of strength $n-1$. However, $G(f)$ has minimum indegree 1, and for every i, $IC(f_i)$ equals 1.

3.1 Graph-based constructions

To obtain k-admissible non-complete graphs, we can use the following results:

Lemma 2. *Let $A \in CA(m_1, n-1; k)$ and $B \in CA(m_2, n-1; k-1)$. Then,*

$$C = \begin{bmatrix} A & \vec{0} \\ B & \vec{1} \end{bmatrix} \in CA(m_1 + m_2, n; k).$$

Proof. Let $I = \{i_1, \dots, i_k\} \subseteq [n]$ and $a = (a_1, \dots, a_k) \in \{0,1\}^k$. Now there are two possible cases. If $n \notin I$ since A is a covering array of strength k, there is a vector $x \in C$ such that $x_I = a$. On the other case $n \in I$, and we write without loss of generality $I = \{i_1, \dots, i_{k-1}, n\}$ and $a = (a_{i_1}, \dots, a_{i_{k-1}}, a_n)$. If $a_n = 0$, since A has strength k there exists $x \in C$ such that $x_I = a$. Otherwise, if $a_n = 1$, as B has strength $k-1$, there is a vector $y \in C$ such that $y_{I \setminus \{n\}} = (a_{i_1}, \dots, a_{i_{k-1}})$, and therefore $y_I = a$. $\square$

Lemma 3. *Let $A \in CA(m_s, n_s; s)$ and $B \in CA(m_r, n_r; r)$. We denote by $A \otimes B$ the set of all possible concatenations between a vector of A and a vector of B:*

$$A \otimes B = \{a_i b_j \in \{0,1\}^{n_s+n_r} \;:\; i, j \in \{1, \dots, s\} \times \{1, \dots, r\}\}.$$

Then, $A \otimes B \in CA(m_s m_r, n_s + n_r; t)$, where $t = \min\{r, s\}$.

Proof. Without loss of generality, assume $t = s$. Let $I = \{i_1, \dots, i_s\} \subseteq [n_s + n_r]$. Consider the partition of I into I_A and I_B, where I_A contains the ℓ_A indices between 1 and n_s, and I_B contains the ℓ_B indices between $n_s + 1$ and $n_s + n_r$. Let $a = a^A a^B \in \{0,1\}^{n_s+n_r}$, where $a^A = \{a_1^A, \dots, a_{\ell_A}^A\}$ and $a^B = \{a_1^B, \dots, a_{\ell_r}^B\}$. Since $t = \min\{s, t\}$, we know that A and B are covering arrays of strength s. Thus, there exist $x \in A$ and $y \in B$ such that $x|_A = a^A$ and $y|_B = a^B$. As $A \otimes B$ contains all possible concatenations of elements between A and B, we conclude that $xy \in A \otimes B$ and, therefore, $A \otimes B \in CA(m_s m_r, n_s + n_r; t)$. □

Corollary 3.3. *Let $\{A^\ell\}_{\ell=1}^L$ be a collection of sets of Boolean vectors such that for every ℓ, A^ℓ is an element of $CA(m_\ell, n_\ell; t_\ell)$. Then,*

$$\bigotimes_{\ell=1}^{L} A^\ell = ((A^1 \otimes A^2) \otimes A^3) \otimes \dots \otimes A^L) \in CA(m, n; t)$$

Where $m = \prod_{\ell=1}^L m_\ell$, $n_\ell = \sum_{\ell=1}^L n_\ell$ and $t = \min\{t_\ell \;\; \ell = 1, \dots, L\}$.

Proposition 4. *Let $\{G_\ell\}_{\ell=1}^L$ be a family of graphs such that for each ℓ, there is a BN f^ℓ such that $G(f^\ell) = G_\ell$ and $i(f^\ell) = t_\ell$. Now we define $G = \bigcup_{\ell=1}^L G_\ell$. Then, there exists a Boolean network f such that $G(f) = G$ and $i(f) = k$, where $k = \min\{t_\ell \;:\; \ell = 1, \dots, L\}$.*

Proof. Since it is a disjoint union, we can define f locally as f^ℓ for each G_ℓ. So, the set of fixed points would be of the form

$$FP(f) = \bigotimes_{\ell=1}^{L} FP(f^\ell)$$

Where each $FP(f^\ell)$ is a covering array of strength t_ℓ. Then, by Lemma 3, we this set is a covering array of strength $k = \min\{t_\ell \;:\; \ell = 1, \dots, L\}$ with $\prod_{\ell=1}^L |FP(f^\ell)|$ elements. □

Example 7. *We can construct a 3-admissible graph with 20 vertices, by the previous proposition, using K_6, K_4, two copies of K_5 and XOR local activation functions.*

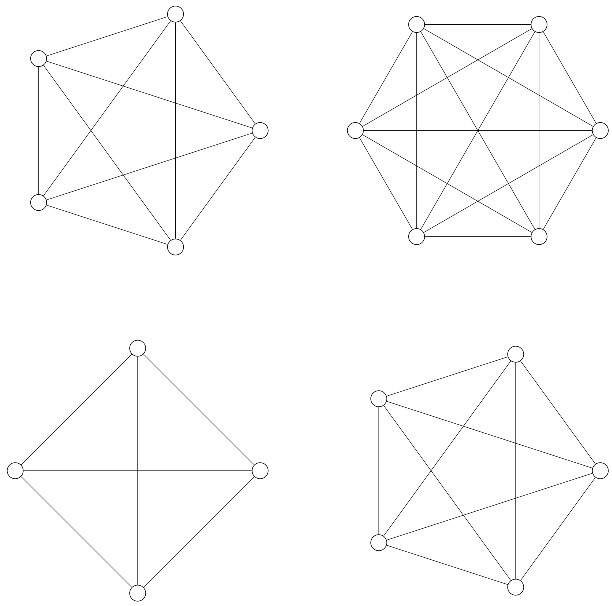


Figure 3.6. A 3-admissible graph with 20 vertices.

There is a particularly interesting case similar to the previous construction. Let's suppose that G is a k-admissible graph on n vertices. What happens if we add an isolated vertex with a loop? For example, we know that $K_5 = G(f)$ is 4-admissible with a XOR network. Then, this new Boolean network $\tilde{f}$ has fixed points

$$FP(\tilde{f}) = \begin{bmatrix} 00000 & 0 \\ 00011 & \vdots \\ 00101 & \vdots \\ 01001 & \vdots \\ \vdots & \vdots \\ 11110 & 0 \\ 00000 & 1 \\ 00011 & \vdots \\ 00101 & \vdots \\ 01001 & \vdots \\ \vdots & \vdots \\ 11110 & 1 \end{bmatrix} = \begin{bmatrix} & 0 \\ & \vdots \\ FP(f) & \vdots \\ & \vdots \\ & 0 \\ & 1 \\ & \vdots \\ FP(f) & \vdots \\ & \vdots \\ & 1 \end{bmatrix}$$

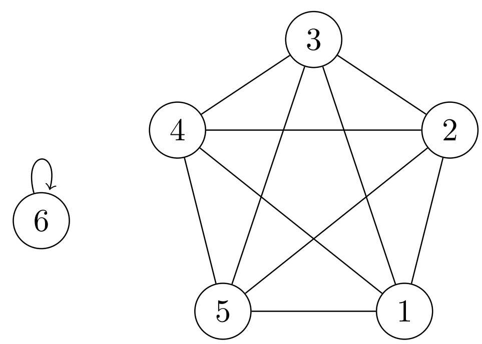


Figure 3.7. K_5 with an extra isolated loop.

Note that for each loop added, the number of fixed points doubles. It is easy to see that $FP(\tilde{f})$ is an element from $CA(6; 4)$. In general, we claim the following result:

Proposition 5. *Let G be a k-admissible graph on n vertices. Then, we can construct a k-admissible graph on $n+1$ vertices.*

Proof. Let $\tilde{G} = G \cup \{n+1\}$ be the graph G with the vertex $n+1$ added as an isolated loop. Let

f be a k-independent BN with interaction graph G. Now we define $\tilde{f} : \{0,1\}^{n+1} \to \{0,1\}^{n+1}$ as $\tilde{f}(x) = (f_1(x), \ldots, f_n(x), x_{n+1})$. So $G(\tilde{f}) = \tilde{G}$, and also

$$FP(\tilde{f}) = \begin{bmatrix} FP(f) & \vec{0} \\ FP(f) & \vec{1} \end{bmatrix}$$

Now, by Lemma 2, $FP(\tilde{f}) \in CA(2FP(f), n+1; k)$. □

In particular, the above construction allows us to use cliques with XOR functions and isolated loops to construct, for any n and k, Boolean networks with $i(f) = k$.

Corollary 3.4. *For any n and $1 < k < n$, there is a non-complete k-admissible graph with n vertices.*

Also, if n is a multiple of k, we can be more precise:

Corollary 3.5. *Let k be an integer and n a multiple of k. Then, there exists a $(k-1)$-regular $(k-1)$-admissible graph on n vertices.*

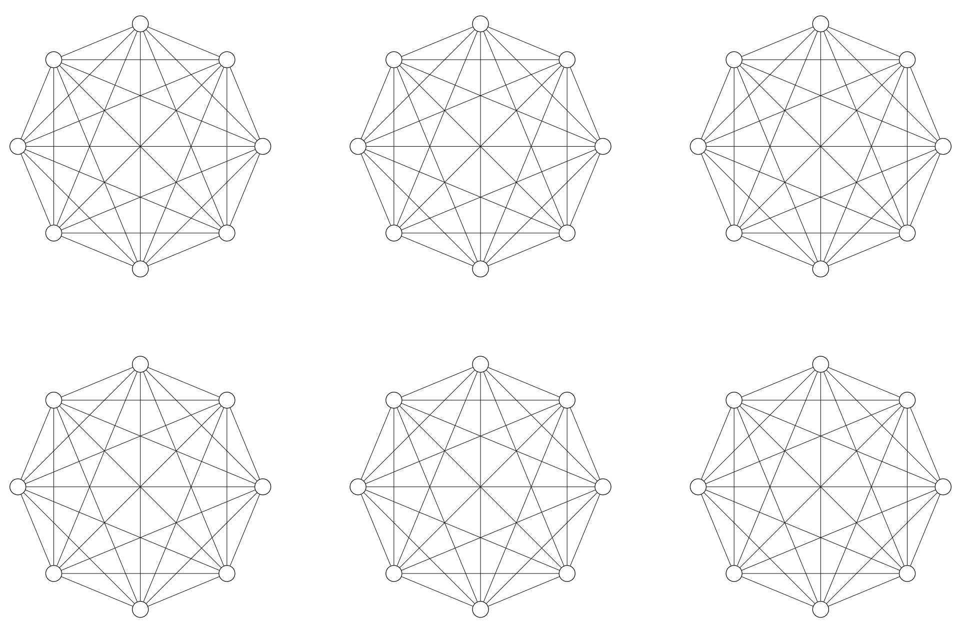


Figure 3.8. A 7-regular 7-admissible graph with 48 vertices.

After recognizing that the inclusion of loops doubles the number of fixed points, we wonder: Can we construct examples of networks with $i(f) = k$ and the maximum number of fixed points without increasing the strength? To advance in this direction, we first prove the following upper bound.

Proposition 6. *Let $A \in CA(n;k) \backslash CA(n;k+1)$. Then, an upper bound for the number of elements of A is*

$$2^{n-1}(2 - 2^{-k})$$

Proof. Since A has no strength $k+1$, there exists $a = (a_1, \ldots, a_{k+1}) \in \{0,1\}^{k+1}$ such that for any vector we select as a completion $b = (b_{k+2}, \ldots, b_n) \in \{0,1\}^{n-k-1}$, the concatenation $ab = (a_1, \ldots, a_{k+1}, b_{k+2}, \ldots, b_n) \in \{0,1\}^n$ is not an element of A. Therefore, there are at least 2^{n-k-1} elements that are not part of the rows of A, so the upper bound is $2^n - 2^{n-k-1} = 2^{n-1}(2-2^{-k})$. □

The following result demonstrates that, for a fixed strength k, we can approach this bound closely (up to a constant).

Proposition 7. *For every $k \leq n$, there is a Boolean network with $i(f) = k$ and 2^{n-1} fixed points.*

Proof. Let us consider a graph G composed by a clique of size $k+1$ and $n-k-1$ isolated loops. Suppose we have a XOR Boolean network with this interaction graph. Then, by the previous results, we know that $i(f) = k$. The inclusion of loops does not increase the strength, as the configuration $\vec{1} \in \{0,1\}^{k+1}$ remains unstable for the isolated clique. Then, since every loop duplicates the set of fixed points, we conclude that f has $2^{n-k-1}2^k = 2^{n-1}$ fixed points. □

Example 8. *Let us consider $n = 8$ and $k = 3$. Then, a covering array of strength 3 and 8 columns has at most $\frac{7}{4}2^7 = 224$ vectors, while the following graph with XOR functions achieves $2^7 = 128$ fixed points.*

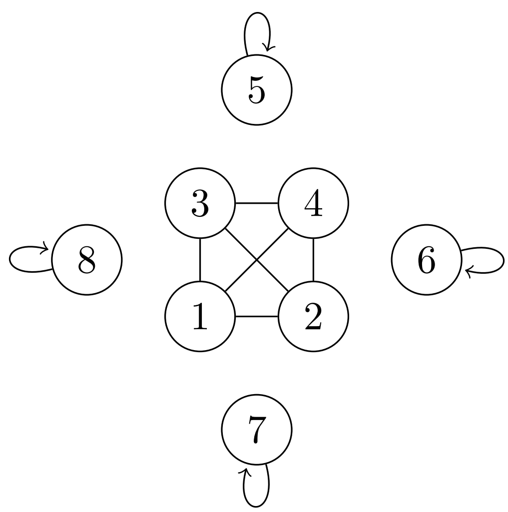


Figure 3.9. Construction from Proposition 7.

Now, the next question we decided to investigate was: Given n, can we construct k-independent networks with a connected interaction graph? The answer was affirmative for certain cases of n and k, as we can see in the following two lemmas:

Lemma 4. *Let r, s be two integers and define $\xi := \min\{r, s\} - 1$. Then, there exists a ξ-admissible connected digraph on $n = r + s$ vertices.*

Proof. Let K_r and K_s denote the cliques on r and s vertices, respectively. Now we define G composed by these two cliques and select $i \in V(K_r)$, and add all the arcs of the form (i, ℓ) for

$\ell \in K_s$. Let $f : \{0,1\}^n \to \{0,1\}^n$ be a XOR Boolean network with $G(f) = G$. Now, we see that for every $x \in FP(f)$, if $x_i = 0$ the number of ones in both cliques should be even. So there are $2^{r-2}2^{s-1}$ fixed points. On the other case, if $x_i = 1$, every vector with an odd number of ones on the variables given by K_s, and an odd number of ones in $K_r \setminus \{i\}$, is a fixed point of f. In this case there are also $2^{r-2}2^{s-1}$ options. In total, there are 2^{r+s-2} fixed points and by previous lemmas this set is a covering array of strength ξ. □

i	$K_r \setminus \{i\}$	K_s
0	even	even
1	odd	odd

Table 3.1. Fixed points of the construction of Lemma 4.

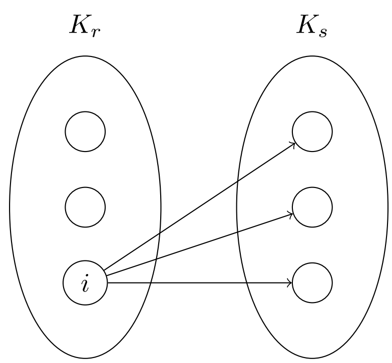


Figure 3.10. $G = K_m \overset{\rightarrow}{\cup} K_n$

We conclude that connectivity does not appear to be a necessary or sufficient condition for the k-admissibility of graphs. Naturally, the question arises as to whether there exist strongly connected k-admissible graphs. Furthermore, we are particularly interested in finding, for a given n and k, a family of strongly connected and k-admissible graphs. We were able to address this question for certain cases of n and k with the following lemma.

Lemma 5. *For any integer $m \geq 2$ and odd $k \geq 1$, there is a strongly connected graph that is $(m-1)$-admissible, with $n = (m-1)k+1$ vertices.*

Proof. We know that cliques achieve high k-independence with XOR functions. Our next construction is built upon this idea. Let $W_{m,k} = (V, E)$ be a graph with $n = (m-1)k+1$ vertices, comprising a central vertex and k copies of K_m, each sharing only the central vertex. Examples of these graphs are shown in Figure 3.11.

We claim that for every m, k with odd k, the XOR Boolean network with interaction graph $W_{m,k}$ is $(m-1)$-independent. To prove this, we will first characterize the set of fixed points of this network. To do so, we denote by f the XOR BN with $G(f) = W_{m,k}$, by 1 the central vertex of the graph, and let $x \in FP(f)$. Now, we distinguish the following two cases:

- If $x_1 = 0$, then we need that the central vertex observes an even number of ones.
- If $x_1 = 1$, then we need for it to observe an odd number of ones.

On the other hand, each of the cliques of size m must have an even number of ones; otherwise, the configuration would be unstable. We denote by $K^1_{m-1}, \dots, K^k_{m-1}$. Then, the set of fixed points of f is given by the configurations that have $x_1 = 0$ and for every $\ell \in \{1, \dots, k\}$, $w_H(x^\ell_{K_{m-1}})$ is even or $x_1 = 1$ and for every $\ell \in \{1, \dots, k\}$, $w_H(x^\ell_{K_{m-1}})$ is odd. Here we note that if k is even, the central vertex cannot take the value 1 on a fixed point, because it will always observe an even number of ones. We can summarize the set of fixed points in the following table:

1	K^1_{m-1}	K^2_{m-1}	$\cdots$	K^k_{m-1}
0	even	even	even	even
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	even	even	even	even
1	odd	odd	odd	odd
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	odd	odd	odd	odd

Table 3.2. Fixed points of $W_{m,k}$ with XOR interaction, k odd.

Considering that for each K^ℓ_{m-1} there are 2^{m-2} possible configurations with even (or odd) weight, we have $2^{(m-2)k}$ fixed points with $x_1 = 0$ and the same amount with $x_1 = 1$. Thus, f has $2^{(m-2)k+1}$ fixed points. Moreover, this set has strength $m-1$. Indeed, let I be a subset of $m-1$ vertices from $W_{m,k}$ and let $a = (a_1, a^{K^{i_1}_{m-1}}, \dots, a^{K^{i_t}_{m-1}}) \in \{0,1\}^{m-1}$, with $t \leq k$. We know that the set of fixed points, for $x_1 = 0$ (or $x_1 = 1$) restricted to any K^ℓ_{m-1} is a covering array of strength $m-2$. Then, there exists a fixed point x such that $x_I = a$, so $FP(f)$ is a covering array of strength $m-1$.

□

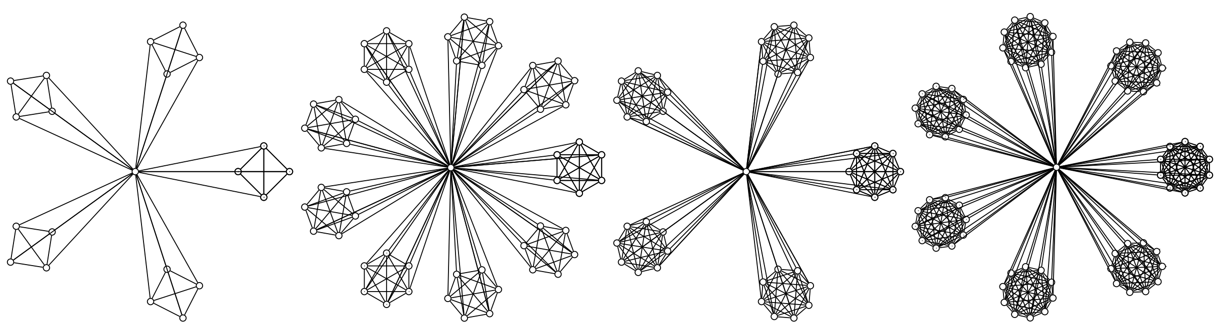


Figure 3.11. Windmill graphs with $(m, k) \in \{(5,5), (7,9), (9,5), (11,7)\}$ (left to right).

Chapter 4

Families of *k*-independent networks

In this section, we will explore various families of k-independent networks for different values of $1 \leq k < n$. We will focus on homogeneous Boolean networks (i.e., all local activation functions are of the same type), delving deeply into three of them: XOR (sum modulo 2), Minority, and Majority. As mentioned earlier, the XOR function potentially leads to the emergence of numerous fixed points, which is why we selected it for this chapter. On the other hand, the motivation behind choosing the other two functions arises entirely from the applied scenario described in the introduction. Since our interest lies in studying k-independence and understanding it within the context of a system of individuals trying to reach a decision, it is natural to assume the existence of friends and enemies. Given a Boolean network f with an undirected interaction graph G, we will say that two adjacent vertices i and j are friends if the function f_i is monotonically increasing to x_i, and vice versa. Conversely, if the local activation function is monotonically decreasing, we will refer to them as enemies. In this context, a network with Maj activation functions represents a case where all interactions are "friendly". On the contrary, the Min function allows modeling a scenario in which all interactions are "unfriendly" (enemies). Additionally, we incorporate the XOR function in the analysis to contrast with the constructions presented in the previous chapter.

Definition 11. *Let* $\mathrm{XOR}, \mathrm{Min}, \mathrm{Maj} : \{0,1\}^n \to \{0,1\}$ *be the Boolean functions defined by:*

$$\begin{aligned} \mathrm{XOR}(x) = 1 &\iff w_H(x) = 1 \mod 2 \\ \mathrm{Min}(x) = 1 &\iff w_H(x) \leq \lfloor n/2 \rfloor \\ \mathrm{Maj}(x) = 1 &\iff w_H(x) \geq \lceil n/2 \rceil \end{aligned}$$

One reason why these functions are interesting for us is due to $IC(\mathrm{XOR}) = n$*,* $IC(\mathrm{Min}) = \lfloor n/2 \rfloor$ *and* $IC(\mathrm{Maj}) = \lceil n/2 \rceil$

Also, we remark that we will explore k-independence from the perspective of functions rather than the interaction graphs. Up to the previous chapter, we have established that employing XOR interactions, combined with various constructions, enables the construction of diverse k-admissible graphs. In this regard, the philosophy of this chapter is to focus on XOR, Min, and Maj for different fixed graphs. We pose the question: What degree of k-independence do different families of graphs allow when considering these functions? The first result is on the complete graph and proves that this is an architecture that allows various degrees of k-independence for different local activation functions.

Proposition 8. *Let* $\mathrm{XOR}, \mathrm{Min}, \mathrm{Maj} : \{0,1\}^n \to \{0,1\}^n$ *denote the three homogeneous Boolean networks with their respective local activation functions and interaction graph* $G = K_n$. *Then,*

$$i(\mathrm{XOR}) = n-1, \quad i(\mathrm{Min}) = \left\lfloor \frac{n}{2} \right\rfloor, \quad i(\mathrm{Maj}) = 1.$$

Proof. The proof for the XOR function is Lemma 1. For the Min network, we claim that

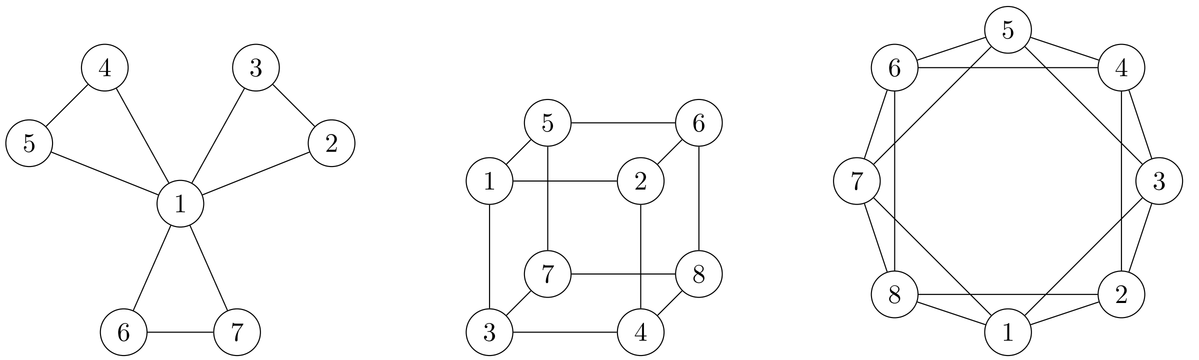
$$FP(\mathrm{Min}) = \left\{ x \in \{0,1\}^n \, : \, w_H(x) = \left\lceil \frac{n}{2} \right\rceil \right\}$$

Indeed, every vector of weight $\lceil n/2 \rceil$ is a fixed point, since every zero will be looking at more ones than zeroes so it will remain as a zero. Similarly, everyone in the vector looks at $\lceil n/2 \rceil - 1$ others. Given the absence of loops (no self-interactions), each one observes more zeroes than ones, resulting in a stable configuration. To see there are no more fixed points, we can suppose by contradiction that $x \in FP(\mathrm{Min})$ with at least $\lfloor n/2 \rfloor + 1$ ones. Consequently, there must be an index $i \in [n]$ such that $x_i = 1$, and it observes at least $\lfloor n/2 \rfloor$ ones. As a result, it will change to zero, presenting a clear contradiction. A similar argument can be used with the case $w_H(x) < \lfloor n/2 \rfloor$.

Now we will prove that $FP(\mathrm{Min})$ has strength $t = \lfloor n/2 \rfloor$. Consider, without loss of generality $I = \{1, \dots, t\} \subseteq [n]$ and $a = (a_1, \dots, a_t) \in \{0,1\}^t$. Then, $w_H(a) \leq t = \lfloor n/2 \rfloor$, so we can concatenate a vector $b \in \{0,1\}^{n-t}$ to a, in order to complete its weight up to t, i.e., with $w_H(b) = \lceil n/2 \rceil - w(a)$. Thus, $ab \in \{0,1\}^n$ and $w(ab) = \lceil n/2 \rceil$, so $ab \in FP(\mathrm{Min})$ and $ab_I = a$. Since I and a are arbitrary, $FP(\mathrm{Min}) \in CA(n;t)$. Also, it is easy to see that this set does not have strength $t+1$ because for every $I \subseteq [n]$ of size $t + 1 = \lceil n/2 \rceil$, the configuration $\vec{0}$ is not possible. This is because there would be $\lfloor n/2 \rfloor$ free positions, making it impossible to achieve a weight of $\lceil n/2 \rceil$, in order to be a fixed point.

Finally, for the Maj function, it is easy to see that $\{\vec{0}, \vec{1}\} \subseteq FP(\mathrm{Maj})$. We will prove that there are no more fixed points. In fact, suppose there exists $x \in FP(\mathrm{Maj})$ and $i, j \in [n]$ such that $x_i = 1$ and $x_j = 0$. We define $J = [n] \setminus i, j$ and emphasize that the component i must observe at least $\lceil n/2 \rceil$ ones from J, while j must observe at least $\lceil n/2 \rceil$ zeroes from J, so J should have at least n elements. This is a contradiction because $|J| = n - 2$.

$\square$



a) A 2-admissible graph with XOR b) A 2-admissible graph with Maj c) A 2-admissible graph with Min

Figure 4.1. Examples of 2-independent BNs with non complete interaction graph.

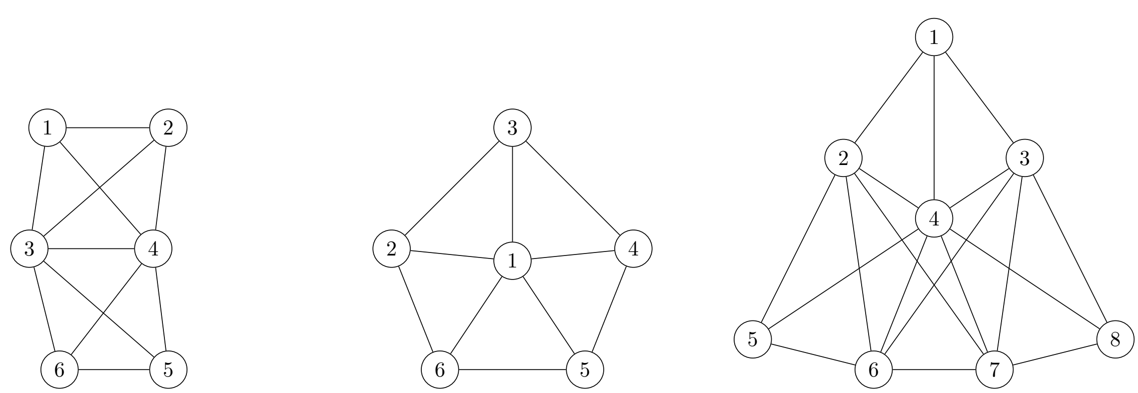
It is worth mentioning that, just like with the XOR function, in the case of Min and Maj, there exist non-complete k-admissible graphs, as illustrated in Figure 4.1.

Before delving into explicit families, let us review a general result that imposes a restriction on $G(f)$ when aiming to achieve a certain k-independence in a homogeneous network f.

Proposition 9. *Let G be a directed loopless graph such that there exist two vertices with the same in-neighborhood of size t. Then there does not exist a homogeneous 2-independent Boolean network f with $G(f) = G$.*

Proof. Suppose, by contradiction, that there is a homogeneous Boolean network f such that $G(f) = G$ and f is 2-independent. Now let $i, j \in V(G)$ such that $N^-(i) = N^-(j) = \{1, \ldots, t\}$ and take $I = N^-(i) \cup \{i, j\}$. Then, the configuration $a = (a_1, \ldots, a_t, a_i, a_j)$ with $a_i = 0$ and $a_j = 1$ is impossible, since the values f_i and f_j are the same. $\square$

Remark 4. *We want to emphasize that, although the XOR function generally allows for a greater number of fixed points, it does not always achieve a higher strength for certain graphs compared to the minority or majority functions. Figure 4.2 shows examples where XOR and Maj are 1-independent while Min is 2-independent.*



#195 #187 #1186

Figure 4.2. Examples of 2-independent BNs with non complete interaction graph, see Table 6.1.

Proposition 10. *For the (m,k)-directed windmill $W_{m,k}$ with $n=(m-1)k+1$ vertices, we denote by* $\mathrm{XOR}, \mathrm{Min}, \mathrm{Maj}$ *the Boolean networks on n variables with interaction graph $W_{m,k}$.*

- *If k is odd, XOR is $(m-1)$-independent.*
- *Min is $\lfloor \frac{m}{2} \rfloor$-independent.*
- *Maj is 1-independent.*

Proof. The XOR case is already proved in Lemma 5. Therefore, we will prove first the Min case. As we did before, we will denote $V(W_{m,k})$ by $1, K^1_{m-1}, \dots, K^k_{m-1}$, where 1 is the central vertex, and for each ℓ, K^ℓ_{m-1} denotes a clique without the central vertex. Let $x \in FP(\mathrm{Min})$, if $x_1 = 0$, we observe that necessarily for each ℓ, $w_H(K^\ell_{m-1}) = \lceil m/2 \rceil$. On the other hand, if $x_1 = 1$, we need $w_H(K^\ell_{m-1}) = \lfloor m/2 \rfloor$ for each ℓ. In summary, the set of points is represented in Table 4.1, where $\lceil m/2 \rceil$ denotes any possible configuration of that weight (similarly for $\lfloor m/2 \rfloor$). A completely analogous argument to the Min case in Proposition 8 can be used to prove that this set has strength $\lfloor m/2 \rfloor$. Moreover, there are $\binom{m-1}{\lfloor \frac{m}{2} \rfloor}^k$ possible vectors $x \in \{0,1\}^n$ with $x_1 = 1$ and $\lfloor \frac{m}{2} \rfloor$ ones in every clique of size $m-1$, while there are $\binom{m-1}{\lceil \frac{m}{2} \rceil}^k$ vectors with $\lceil \frac{m}{2} \rceil$ ones in every clique of size $m-1$.

1	K^1_{m-1}	K^2_{m-1}	$\cdots$	K^k_{m-1}
0	$\lceil m/2 \rceil$	$\lceil m/2 \rceil$	$\lceil m/2 \rceil$	$\lceil m/2 \rceil$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	$\lceil m/2 \rceil$	$\lceil m/2 \rceil$	$\lceil m/2 \rceil$	$\lceil m/2 \rceil$
1	$\lfloor m/2 \rfloor$	$\lfloor m/2 \rfloor$	$\lfloor m/2 \rfloor$	$\lfloor m/2 \rfloor$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	$\lfloor m/2 \rfloor$	$\lfloor m/2 \rfloor$	$\lfloor m/2 \rfloor$	$\lfloor m/2 \rfloor$

Table 4.1. Fixed points of $W_{m,k}$ with Min interaction.

Finally, for the Maj case, we just need to remember that, for the same reason as Proposition 8, in every fixed point of Maj, each set of indexes K^ℓ_{m-1} is either $\vec{0}$ or $\vec{1}$.

$\square$

Proposition 11. *Consider two Boolean networks denoted as $f, \tilde{f} : \{0,1\}^n \to \{0,1\}^n$, such that f is k-independent and $\tilde{f}$ is $(k-1)$-independent. Additionally, let $g : \{0,1\}^{n+1} \to \{0,1\}^{n+1}$ be defined by*

$$g_i(x) = (x_{n+1} \wedge \tilde{f}_i(x)) \vee (\overline{x_{n+1}} \wedge f_i(x)), \;\; i \in \{1,\dots,n\},$$

and $g_{n+1}(x) = x_{n+1}$. Then, it follows that g is k-independent.

Proof. Let us note that if $x_{n+1} = 0$, then $g(x) = f(x)$, while if $x_{n+1} = 1$, $g(x) = \tilde{f}(x)$. So, the set of fixed points of g is

$$FP(g) = \begin{bmatrix} FP(f) & \vec{0} \\ FP(\tilde{f}) & \vec{1} \end{bmatrix}$$

And by Lemma 2, $FP(g) \in CA(n+1;k)$ and therefore $i(g) \geq k$. $\square$

Remark 5. *We can also state the previous proposition in the following manner: Given G^1, G^2 to graphs on $V = [n]$, such that G^1 is k-admissible and G^2 is $(k-1)$-admissible, then we can construct $\tilde{G} = (\tilde{V}, \tilde{E})$, where $\tilde{V} = [n+1]$ and $\tilde{E} = E(G^1) \cup E(G^2)$. Thus, by the previous proposition, we can define the same network and conclude that $\tilde{G}$ is a k-admissible graph on $n+1$ vertices. In Figure 4.3, we observe an example of this construction considering the Maj network in G^1, being 2-independent, and the XOR network in G^2 achieving 3-independence. In this case, $\tilde{G}$ is the resulting graph, which turns out to be 3-admissible with the network defined in Proposition 11.*

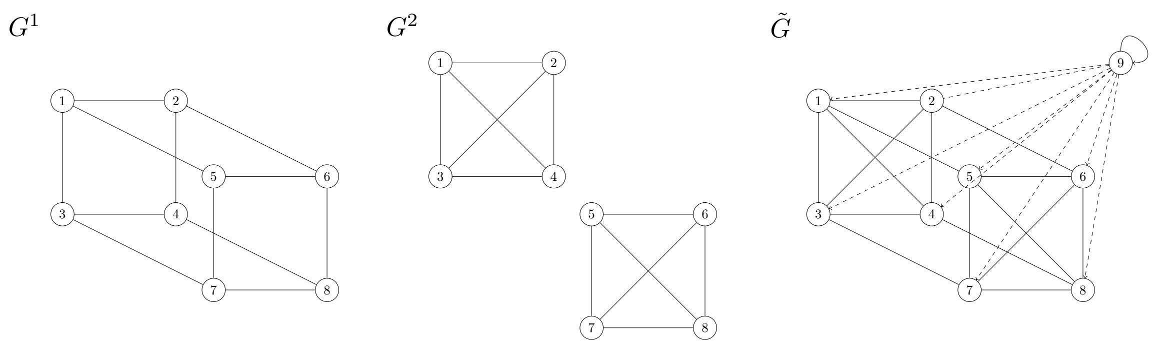


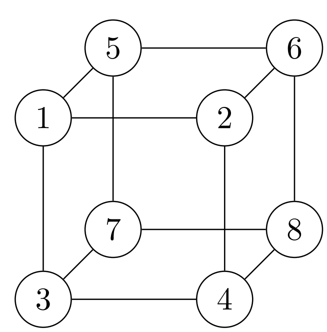
Figure 4.3. Construction from Proposition 11 using G^1 with Maj and G^2 with XOR.

4.1 The monotone case

In order to find more examples of k-independent networks, the decision was made to use the "Graph Atlas" database [25] available in the Python library Networkx. The exhaustive search, with graphs up to 7 vertices, did not yield in any k-admissible graphs for the Majority function, for any $k \geq 2$, as we can see in 6.1. This led us to initially conjecture that k-independence is a property somewhat incompatible with monotone functions. However, this reasoning was false, since we found out that Q_3 is 2-admissible with Maj interaction, as we can see in the following example.

Example 9. *The Maj Boolean network with interaction graph Q_3 has fixed points*

$$FP(\text{Maj}) = \begin{matrix} 00000000 \\ 00001111 \\ 00110011 \\ 01010101 \\ 10101010 \\ 11001100 \\ 11110000 \\ 11111111 \end{matrix} \in CA(8,8;2).$$



As mentioned earlier, there is a vast literature regarding covering arrays, generalizations, and related combinatorial designs. The quest for monotonous k-independent networks, with $k \geq 2$, led us to discover the Steiner systems, which as we will see later, are structures that under certain conditions form covering arrays, and in turn allow a representation as fixed points of a monotone network.

Definition 12. *Let $A = \{x^1, \ldots, x^m\} \subseteq \{0,1\}^n$. We say that A is a Steiner system with parameters (n,k,t) if $w_H(x^i) = k$ for $i = 1, \ldots, m$, and for every subset of indices $I = \{i_1, \ldots, i_t\}$ there is an unique vector $x^j \in A$ such that $x^j_{i_\ell} = 1$ for $\ell \in \{1, \ldots, t\}$.*

Example 10. *The following is a Steiner system with parameters $(8,4,3)$:*

$$A = \begin{matrix} 11010001 \\ 01101001 \\ 00110101 \\ 00011011 \\ 10001101 \\ 01000111 \\ 10100011 \\ 00101110 \\ 10010110 \\ 11001010 \\ 11100100 \\ 01110010 \\ 10111000 \\ 01011100 \end{matrix}$$

The existence of a Steiner system with given parameters is an old problem in combinatorics [9]. In a more general context, divisibility conditions were deduced: for the existence of a (n,q,r) Steiner system, a necessary condition is that $\binom{q-i}{r-i}$ divides $\binom{n-i}{r-i}$ for every $0 \leq i \leq r-1$. For many years, it was conjectured that the divisibility conditions were also sufficient, and this was proved in 2014, for large values of n [17].

Lemma 6. *Let A be a Steiner system with parameters $(n, t+1, t)$ such that $2t < n$. Then, A is a covering array of strength t.*

Proof. Let I be a subset of $[n]$ of size t, we will assume without loss of generality that $I = \{1, \ldots, t\}$. We aim to prove that for every $a = (a_1, \ldots, a_t) \in \{0,1\}^t$, there exists $x \in A$ such that $x_I = a$. We will proceed with the proof by induction on the number of zeros in a.

First, observe that there exists $x^\ell \in A$ such that $x^\ell_I = 11\cdots1$, due to the property of Steiner systems. As the vectors have weight $t+1$, there exists a unique $i_0 \in t+1, \ldots, n$ such that $x^\ell_{i_0} = 1$. Now, let $\overline{e_{i_0}} \in \{0,1\}^t$ be the vector that has only one zero in position i_0 and $J = \{j_1, \ldots, j_{t-1}\} \in \{t+1, \ldots, n\}$. Consider $w \neq i_0$ and define $K^{i_0} = (\{1, \ldots, t\} \setminus \{i_0\}) \cup \{w\}$. Notice K^{i_0} is a subset of t indices, so there exists a vector x^{ℓ_1} that has ones in the components K^{i_0}. Suppose $x^{\ell_1}_{i_0} = 1$. In such case, the configuration $\overline{e_{i_0}}$ in the indices I would have two distinct completions (since $w \neq i_0$).

Therefore, $x^{\ell_1}_{i_0}$ must be zero, and hence $x^{\ell_1}_I = \overline{e_{i_0}}$. With this, we proved that given a subset of t indices, all configurations with one zero and $t-1$ ones appear.

Now, suppose that all configurations with s zeros appear, and let us prove that those with $s+1$ zeros also appear. Let $a = (a_1, \dots, a_t) \in \{0,1\}^t$ such that $a_1 = \cdots = a_{s+1} = 0$ and $a_{s+2} = \cdots a_t = 1$. We will prove that there exists an element of the Steiner system that takes the values of a at the indices I. Consider the vector x^s that completes the configuration $b = (b_1, \dots, b_t)$ with values $b_1 = \cdots = b_s = 0$, $b_{s+1} = \cdots = b_t = 1$ (which exists by the induction hypothesis). Now, let $J = \{\ell \in t+1, \dots, n \,:\, x^s_\ell = 1\}$. As the vectors of the Steiner system have weight $t+1$, $|J| = s+1$. We denote $J = \{j_1, \dots, j_s, j_{s+1}\}$, and consider $w \in \{t+1, \dots, n\} \setminus J$, which allows us to define $K^i = (\{j_1, \dots, j_s\} \cup \{w\}) \cup \{s+2, \dots, t\}$, which is a subset of $[n]$ of size t, so there exists $y \in A$ that takes the value one in the components indexed by K^i, and also has another component with value one. Note that if $y_{s+1} = 1$, we would have two different completions for $\{s+1, \dots, t\} \cup J \setminus \{j_{s+1}\}$, which is a contradiction. Now, if there exists $\ell \in \{1, \dots, s\}$ such that $y_\ell = 1$, we can consider, instead of x^s, the vector ξ^s such that $\xi^s_\ell = 1$, $\xi^s_{s+2} = \cdots = \xi^s_t = 1$, and define J based on ξ^s, and thus repeat the same argument as before. We thus conclude that there must exist $\zeta \in \{t+1, \dots, n\} \setminus K^i$ such that $y_\zeta = 1$, and therefore $y_I = a$.

□

Theorem 4. *Given a Steiner system A with parameters $(n, t+1, t)$, with $t \geq 2$, there is a t-independent monotone Boolean network f whose fixed points contain A and $G(f)$ is the complete loopless graph.*

Proof. Let $A = \{y^1, \dots, y^m\}$ be a $(n, t+1, t)$-Steiner system. By the previous lemma, we know that A is a covering array of strength t. Now for every $i \in [n]$ we define the Boolean function

$$f_i(x_1, \dots, x_n) = \bigvee_{\{k \,:\, y^k_i = 1\}} \bigwedge_{\{j \neq i \,:\, y^k_j = 1\}} x_j.$$

Now we will prove that $A \cup \{\vec{0}, \vec{1}\} \subseteq FP$. Indeed, it is clear that $\vec{0}$ and $\vec{1}$ are fixed points of f. Let $y^\ell \in A$, and let us prove that $f(y^\ell) = y^\ell$. Let $i \in [n]$, and suppose initially that $y^\ell_i = 0$. By contradiction, suppose $f_i(y^\ell) = 1$, and therefore there exists $k \in [m]$ where $y^k_i = 1$ and for every $j \neq i$ such that $y^k_j = 1$, we have that $y^\ell_j = 1$. Notice that the above would imply that the index set $I = \{j \neq i \,:\, y^k_j = 1\}$, which has size t, has two different completions, one by y^ℓ and the other by y^k. This contradicts the uniqueness of the definition of Steiner systems. On the other hand, suppose now that $y^\ell_i = 1$. In this case, within the expression for $f_i(y^\ell)$, the following conjunction appears:

$$\bigwedge_{\{j \neq i \,:\, y^\ell_j = 1\}} y^\ell_j$$

Therefore, $f_i(y^\ell) = 1$. This implies that for any y^ℓ in A, $f(y^\ell) = y^\ell$, which is equivalent to $A \subseteq FP(f)$.

Now we will prove that $G(f) = K_n$. To do this, we first notice that since f_i can be written as a DNF formula without negated variables, f_i depends on the variable x_j if it appears in any clause. That is, (j, i) is an arc in $G(f)$ if and only if there exists $y^k \in A$ such that $y^k_i = 1$ and $y^k_j = 1$, with $j \neq i$. Indeed, if $i \neq j \in [n]$, then we can consider any completion $T \subseteq [n] \setminus \{i, j\}$ with $|T| = t-2$. Then, by considering $T \cup \{i, j\}$, we have a subset of t indices in $[n]$, and by definition, there exists a unique $y^k \in A \subseteq FP(f)$ such that $y^k_i = y^k_j$ and $y_T = \vec{1}$. Therefore, $(j, i) \in G(f)$, and as these are two arbitrary vertices, we conclude that $G(f) = K_n$.

□

Remark 6. *Example 9 shows that there exists a k-independent monotone Boolean network, whose set of fixed points is not a Steiner system with parameters $(n, k+1, k)$.*

Example 11. *The set $A = \{1101000, 0110100, 0011010, 0001101, 1000110, 0100011, 1010001\}$ is a Steiner system with parameters $(7, 3, 2)$. The previous construction gives us the network*

$$\begin{aligned}
f_1(x) &= (x_2 \wedge x_4) \vee (x_5 \wedge x_6) \vee (x_3 \wedge x_7)\\
f_2(x) &= (x_1 \wedge x_4) \vee (x_3 \wedge x_5) \vee (x_6 \wedge x_7)\\
f_3(x) &= (x_2 \wedge x_5) \vee (x_4 \wedge x_6) \vee (x_1 \wedge x_7)\\
f_4(x) &= (x_1 \wedge x_2) \vee (x_3 \wedge x_6) \vee (x_5 \wedge x_7)\\
f_5(x) &= (x_2 \wedge x_3) \vee (x_4 \wedge x_7) \vee (x_1 \wedge x_6)\\
f_6(x) &= (x_3 \wedge x_4) \vee (x_1 \wedge x_5) \vee (x_2 \wedge x_7)\\
f_7(x) &= (x_4 \wedge x_5) \vee (x_2 \wedge x_6) \vee (x_1 \wedge x_3)
\end{aligned}$$

Example 12. *In 1908, Barrau [5] proved the uniqueness of the $(8, 4, 2)$ Steiner system described by the matrix A:*

$$A = \begin{matrix}
11010001\\
01101001\\
00110101\\
00011011\\
10001101\\
01000111\\
10100011\\
00101110\\
10010110\\
11001010\\
11100100\\
01110010\\
10111000\\
01011100
\end{matrix}$$

Using this we can construct the following monotone Boolean network with $i(f) = 3$:

$$\begin{aligned}
f_1(x) &= (x_2 \wedge x_4 \wedge x_8) \vee (x_5 \wedge x_6 \wedge x_8) \vee (x_3 \wedge x_7 \wedge x_8) \vee (x_4 \wedge x_6 \wedge x_7) \vee (x_2 \wedge x_5 \wedge x_7) \\
&\quad \vee (x_2 \wedge x_3 \wedge x_6) \vee (x_3 \wedge x_4 \wedge x_5) \\
f_2(x) &= (x_1 \wedge x_4 \wedge x_8) \vee (x_3 \wedge x_5 \wedge x_8) \vee (x_6 \wedge x_7 \wedge x_8) \vee (x_1 \wedge x_5 \wedge x_7) \vee (x_1 \wedge x_3 \wedge x_6) \\
&\quad \vee (x_3 \wedge x_4 \wedge x_7) \vee (x_4 \wedge x_5 \wedge x_6) \\
f_3(x) &= (x_2 \wedge x_5 \wedge x_8) \vee (x_4 \wedge x_6 \wedge x_8) \vee (x_1 \wedge x_7 \wedge x_8) \vee (x_5 \wedge x_6 \wedge x_7) \vee (x_1 \wedge x_2 \wedge x_6) \\
&\quad \vee (x_2 \wedge x_4 \wedge x_7) \vee (x_1 \wedge x_4 \wedge x_5) \\
f_4(x) &= (x_1 \wedge x_2 \wedge x_8) \vee (x_3 \wedge x_6 \wedge x_8) \vee (x_5 \wedge x_7 \wedge x_8) \vee (x_1 \wedge x_6 \wedge x_7) \vee (x_2 \wedge x_3 \wedge x_7) \\
&\quad \vee (x_1 \wedge x_3 \wedge x_5) \vee (x_2 \wedge x_5 \wedge x_6) \\
f_5(x) &= (x_2 \wedge x_3 \wedge x_8) \vee (x_4 \wedge x_7 \wedge x_8) \vee (x_1 \wedge x_6 \wedge x_8) \vee (x_3 \wedge x_6 \wedge x_7) \vee (x_1 \wedge x_2 \wedge x_7) \\
&\quad \vee (x_1 \wedge x_3 \wedge x_4) \vee (x_2 \wedge x_4 \wedge x_6) \\
f_6(x) &= (x_3 \wedge x_4 \wedge x_8) \vee (x_1 \wedge x_5 \wedge x_8) \vee (x_2 \wedge x_7 \wedge x_8) \vee (x_3 \wedge x_5 \wedge x_7) \vee (x_1 \wedge x_4 \wedge x_7) \\
&\quad \vee (x_1 \wedge x_2 \wedge x_5) \vee (x_2 \wedge x_3 \wedge x_4) \\
f_7(x) &= (x_4 \wedge x_5 \wedge x_8) \vee (x_2 \wedge x_6 \wedge x_8) \vee (x_1 \wedge x_3 \wedge x_8) \vee (x_3 \wedge x_5 \wedge x_6) \vee (x_1 \wedge x_4 \wedge x_6) \\
&\quad \vee (x_1 \wedge x_2 \wedge x_5) \vee (x_2 \wedge x_3 \wedge x_4) \\
f_8(x) &= (x_1 \wedge x_2 \wedge x_4) \vee (x_2 \wedge x_3 \wedge x_5) \vee (x_3 \wedge x_4 \wedge x_6) \vee (x_4 \wedge x_5 \wedge x_7) \vee (x_1 \wedge x_5 \wedge x_6) \\
&\quad \vee (x_2 \wedge x_6 \wedge x_7) \vee (x_1 \wedge x_3 \wedge x_7).
\end{aligned}$$

Chapter 5

Conclusions and Future work

In this thesis, we introduced a new parameter to define families of Boolean networks based on the structure of their set of fixed points, rather than just their quantity. In this sense, the introduction of the concept of k-independence provides a tool for identifying graphs with a high number of fixed points and a particular structure. Moreover, the perspective from k-admissibility allows us to extend these concepts to generate a family of graphs that behave well when seeking k-independence.

Regarding the achievement of the objectives of this research, we have made significant progress in understanding the existence of k-independent networks for a given k. Furthermore, we have identified necessary conditions in terms of the interaction graph for k-independence in Boolean networks. However, given the difficulty in finding patterns to elucidate a sufficient condition, we decided to restrict ourselves to the XOR, Min, and Maj functions. From this restriction, we were able to infer the existence of monotone k-independent networks based on Steiner systems.

It is notable that the attainment of k-independence does not necessarily mandate a high degree of symmetry, contrary to intuitive expectations. For instance, in the context of the windmill Lemma 5, a central vertex assumes the role of a coordinator to facilitate k-independence. Likewise, the presence of a clique of size k as a subgraph is not an obligatory condition, as exemplified by the case of the 3-cube. Nevertheless, these observations retain significance within the framework of binary decision systems delineated in the introduction. They imply, for instance, that the establishment of k-independent coordination does not inherently hinge upon the existence of a fully connected group of k individuals.

For the applied case, we could understand that an individual represents a vertex without a loop within the network if they decide their opinion solely based on the opinions of their friends, without taking into account their previous opinion. On the other hand, an individual represents a vertex with a loop if they always consider their previous opinion in the decision-making process. It is debatable if loops are relevant for the applied situation. We will refer to nodes with loops as "self-reliant" individuals. We can translate the necessary conditions for a k-independent network in this context. Thus, corollary 3.1 implies that each individual within a k-independent network must consider the opinion of at least k other individuals, provided that none of them are "self-reliant". On the other hand, Theorem 3 indicates that k-independence is heavily influenced by how individuals make decisions within these systems; there cannot be an individual whose opinion is automatically influenced by fewer than k individuals; they must always consider at least k individuals (again, without "self-reliant" individuals). Furthermore, how individuals make decisions is crucial to achieving k-independence, as the same graph (i.e., the same configuration of individuals) can show three different levels of k-independence depending on the decision-making functions used. For example, considering the complete graph, where there are no loops and each individual considers the opinion of all other $n-1$ individuals, if each individual follows the majority opinion, only a 1-independence is achieved. On the other hand, if each individual follows the minority opinion, we proved that the system is $\lfloor n/2 \rfloor$-independent. Additionally, considering the XOR function, which may not sound as natural in applied terms, achieves $(n-1)$-independence. In particular, the example of the complete graph was interesting because it showed us that the majority function, which is perhaps the most natural one in decision-making, did not achieve a high number of independence. This was further reinforced after conducting an exhaustive search for examples. Therefore, we finally wondered if monotonic networks, i.e., decision-making systems in which all interactions are "friendly", could exhibit an arbitrary number of k-independence, and we were able to resolve this through the use of Steiner systems. Consequently, we assert that these findings possess considerable applicability to real-world problems and, therefore, represent an initial step towards leveraging this concept in applied research.

There are several directions for further research on this issue. One natural area of inquiry would be to understand the nature of Boolean functions h with $IC(h) \geq k$. Solving this would enable the identification of families that could potentially serve as local activation functions for k-independent networks. For these families, it would be valuable to explore whether there are sufficient conditions that can be imposed on the interaction graph for them to be k-admissible. Additionally, questions regarding complexity remain open, as we have chosen to focus on other topics within the scope of this undergraduate thesis, and many complexity problems in covering arrays are still unresolved.

On the other hand, it is noteworthy that the definition of k-independence presented in this work differs from the VC-dimension notion studied in [24] in the sense that we now demand that for any subset of k indices and assigned values, there exists a fixed point whose projection onto this set of indices coincides with those values. In contrast, VC-dimension requires the existence of a subset of size k with this property. As a natural extension of this idea, we propose considering Covering Arrays on graphs [21] or Covering Arrays avoiding Forbidden Edges [8], where certain subsets of size k of indices must satisfy this aforementioned property.

Chapter 6

Appendix

6.1 Computational elements.

For the computational simulations conducted, we used the following Python libraries.

```
import os
import pandas as pd
from sympy import *
import itertools
import random
import networkx as nx
import matplotlib.pyplot as plt
```

We also defined the following functions, which proved to be versatile utilities for experimenting with k-independent networks.

Listing 6.1. Function to generate a list with every element from $\{0,1\}^n$.

```
def get_Bk_2(n):
    b_N = []
    for i in range(2**n):
        value = bin(i)[2:].zfill(n)
        b_N.append([int(elem) for elem in value])
    return b_N
```

Listing 6.2. Function to generate a random Boolean vector of n variables as a string.

```
def rand_vec(n):
    key1 = ""
    for i in range(n):
        temp = str(random.randint(0, 1))
        key1 += temp
    return(key1)
```

Listing 6.3. Function that generates every subset from 1,..., n of size k.

```python
def find_subsets(n, k):
    set_n = set(range(n))
    return list(itertools.combinations(set_n, k))
```

Listing 6.4. Function that gives 1 if the projection of a into L is equal to v.

```python
def compare_indxed_vec(a,v, L):
    new_a = []
    for i in L:
        new_a.append(a[i])
    if new_a == v:
        return True
    else:
        return False
```

Listing 6.5. This function computes a clause associated to a vector in a dnf or cnf formula.

```python
def get_clause_expression(values_string, truth_value):
    expression = '('
    for k in range(len(values_string)):
        if truth_value == '1':
            if values_string[k] == '1':
                expression = expression + 'x'+str(k+1) + '&'
            else:
                expression = expression + '!x' +str(k+1) + '&'
        elif truth_value == '0':
            if values_string[k] == '0':
                expression = expression + 'x'+str(k+1) + '|'
            else:
                expression = expression + '!x' +str(k+1) + '|'
    return(expression[:-1]+')')
```

Listing 6.6. This function computes a dnf-formula from the truth table.

```python
def get_local_activation_expression(X, Fx, j):
    local_exp = str()
    for i in range(len(X)):
        if Fx[i][j] == '1':
            local_exp = local_exp+get_clause_expression(X[i],'1') + '|'
    return(local_exp[:-1])
```

Listing 6.7. This function generates a random Boolean vector as a list.

```
def rand_vec_vec(n):
    key1 = ""
    for i in range(n):
        temp = str(random.randint(0, 1))
        key1 += temp
    return([int(elem) for elem in key1])
```

Listing 6.8. This function generates an array of m random Boolean vectors of n variables.

```
def gen_random_vec(m, n):
    if m>2**n:
        return False
    temp_set = set()
    while len(temp_set)<m:
        vector = rand_vec(n)
        temp_set.add(vector)
    temp_list = list(temp_set)
    output = [[int(elem) for elem in temp_list[i]] for i in range
       (len(temp_list))]
    return output
```

Listing 6.9. This function checks whether the set A is a covering array of strength k.

```
def is_CA_vec(A,k):
    n = len(A[0])
    if n<k:
        return False
    for I in find_subsets(n,k):
        for v in get_Bk_2(k):
            val = False
            for a in A:
                 if compare_indxed_vec(a,v,I) == True:
                        val = True
            if val == False:
                return False
    return val
```

Listing 6.10. This function computes the maximum strength of the set A.

```
def maximumstrength(A):
  if len(A)== 0:
    return(0)
  cA = [list(a) for a in A]
  actual_best = 0
  for i in range(1,len(A[0])+1):
```

```
    if is_CA_vec(cA, i):
      actual_best = i
    else:
      return(actual_best)
  return(actual_best)
```

Listing 6.11. This function provides, when feasible, a covering array of strength k comprising m vectors with n variables.

```
def gen_one_CA_vec(m,n,k):
    isCA = False
    while isCA == False:
        random_matrix = gen_random_vec(m,n)
        if is_CA_vec(random_matrix, k):
            isCA = True
    return random_matrix
```

Listing 6.12. This function removes a fixed number of elements from a set randomly.

```
def remove_random_rows(CA, num):
    new_array = CA.copy()
    total_rows = len(new_array)
    indx_rows_to_remove = reversed(sorted(random.sample(range(
       total_rows),num)))
    for row_indx in indx_rows_to_remove:
        del new_array[row_indx]
    return(new_array)
```

Listing 6.13. This function tries to reduce the amount of rows of a Covering array, by removing random rows and keeping the strength.

```
def reduceCA(CA,k, maxiter):
    min_actual = CA.copy()
    for t in range(len(CA)-2**k):
        for iteration in range(maxiter):
            new_CA = remove_random_rows(CA,t)
            if is_CA_vec(new_CA, k):
                print('Lo logramos reducir a ', len(new_CA), '
                   filas')
                min_actual = new_CA
                break
    return(min_actual)
```

Listing 6.14. This function takes a list of fixed points, FixP, and generates a random Boolean network that only has these fixed points.

```
def random_complete_withoutmoreFP(FixP):
    n = len(FixP[0])
    X = [tuple(vector) for vector in get_Bk_2(n)]
    IncompleteTable = dict.fromkeys(X, None)
    for vector in IncompleteTable:
        for element in FixP:
            if vector == tuple(element):
                IncompleteTable[vector] = vector
    a = tuple(rand_vec_vec(n))
    for vector in IncompleteTable:
        if IncompleteTable[vector] == None:
            a = tuple(rand_vec_vec(n))
            while a == vector:
                a = tuple(rand_vec_vec(n))
            IncompleteTable[vector] = a
    exprs = []
    variables = ['x'+ str(i) for i in range(1,n+1)]
    for j in range(n):
        temp_mat = []
        for element in IncompleteTable.keys():
            if IncompleteTable[element][j] == 1:
                temp_mat.append(element)
        boolean_expr = str(SOPform(variables, temp_mat, [])).
           replace('~','!')
        exprs.append(boolean_expr)
    return(IncompleteTable, exprs)
```

Listing 6.15. Belt, Xor, Min and Maj functions.

```
def belt(x,k):
  n = len(x)
  zeros = tuple([0]*n)
  weight = distance(x, zeros)
  if weight == k:
    return(1)
  else:
    return(0)
def xor(x):
  n = len(x)
  zeros = tuple([0]*n)
  weight = distance(x, zeros)
  return(int(weight%2))

```

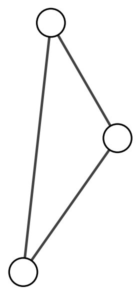
```python
def majo(x):
  n = len(x)
  zeros = tuple([0]*n)
  weight = distance(x, zeros)
  amount_of_zeros = n - weight
  if amount_of_zeros > weight:
    return(0)
  else:
    return(1)

def mino(x):
  n = len(x)
  zeros = tuple([0]*n)
  weight = distance(x, zeros)
  amount_of_zeros = n - weight
  if amount_of_zeros < weight:
    return(0)
  else:
    return(1)
```

```python
diccionario_con_todos_los_grafos = dict()
for i in range(1252):
  G = nx.graph_atlas(i+1)
  n = len(G)
  Bn = [tuple(a) for a in get_Bk_2(n)]
  relabel = dict(zip(G.nodes(), list(range(len(G)))))
  G = nx.relabel_nodes(G, relabel)
  D = {node: list(G[node]) for node in G}
  Fxor = build_from_graph_and_functions(D,xor)
  Fmin = build_from_graph_and_functions(D,mino)
  Fmajo = build_from_graph_and_functions(D,majo)
  fpxor = [x for x, y in zip(Bn, Fxor) if x == y]
  fpmin = [x for x, y in zip(Bn, Fmin) if x == y]
  fpmajo =[x for x, y in zip(Bn, Fmajo) if x == y]
  temp_tuple =  maximumstrength([list(a) for a in fpxor]),
     maximumstrength([list(a) for a in fpmin]), maximumstrength([
     list(a) for a in fpmajo])
  diccionario_con_todos_los_grafos[i+1] = temp_tuple
  print(i+1, 'xor, min, maj: ',temp_tuple)
```

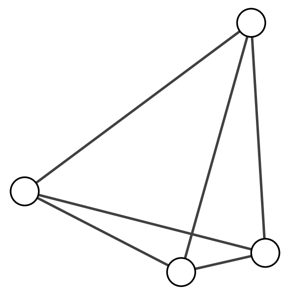
Listing 6.16. Code to iterate over atlas of graphs.

6.2 *k*-admissible graphs on at most 7 vertices.

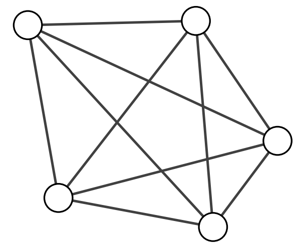
The following graphs are part of the 1252 graphs with $n \leq 7$ vertices listed in [25]. We specifically chose these 20 graphs because for each of them, at least one of the three functions (XOR, Min, Maj) achieves an independence number greater than or equal to two.



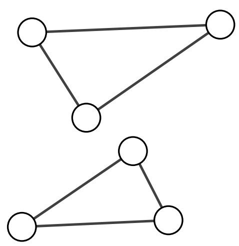
Graph $\#7$.



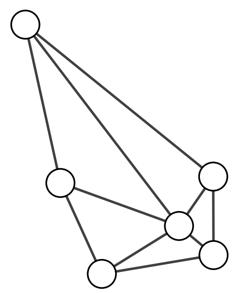
Graph $\#18$.



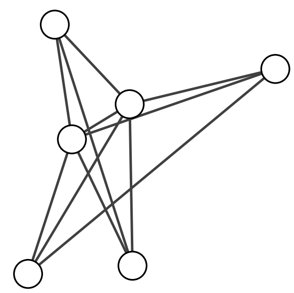
Graph $\#52$.



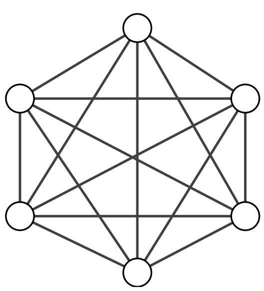
Graph $\#106$.



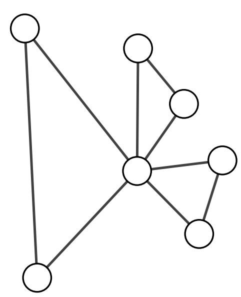
Graph $\#187$.



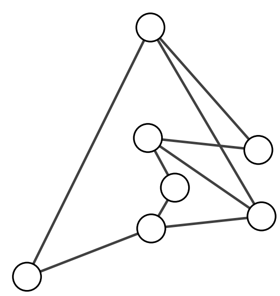
Graph $\#195$.



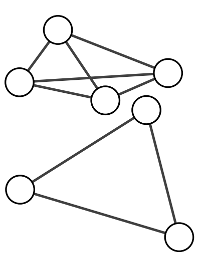
Graph $\#208$.



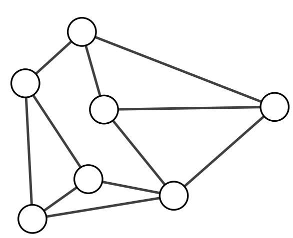
Graph $\#551$.



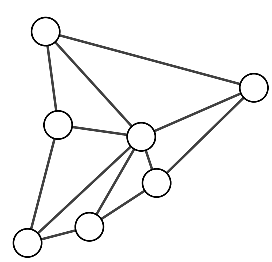
Graph $\#581$.



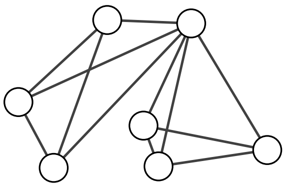
Graph $\#582$.



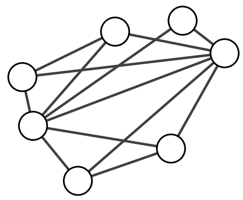
Graph $\#878$.



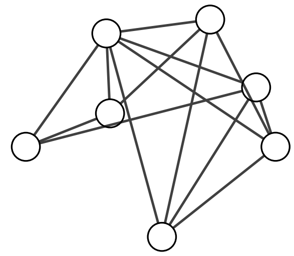
Graph $\#1008$.



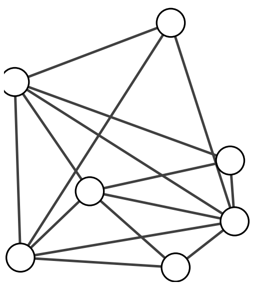
Graph $\#1009$.



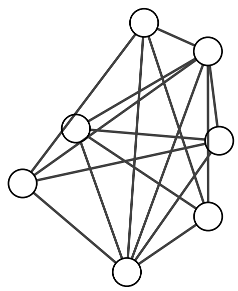
Graph $\#1038$.



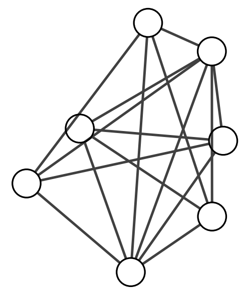
Graph $\#1151$.



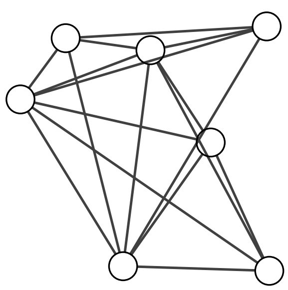
Graph $\#1186$.



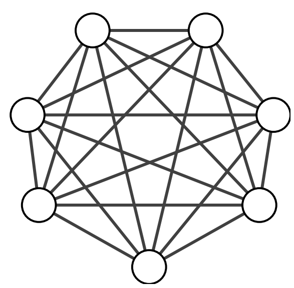
Graph $\#1188$.



Graph $\#1227$.



Graph $\#1238$.



Graph $\#1252$.

Figure 6.2. Some graphs following the enumeration by [25].

Maximum k-independency			
Graphs from Fig. 6.2	XOR	Min	Maj
7	2	1	1
18	3	2	1
52	4	2	1
106	2	1	1
187	1	2	1
195	1	2	1
208	5	3	1
551	2	1	1
581	2	1	1
582	2	1	1
878	0	2	1
1008	0	2	1
1009	0	2	1
1038	2	1	1
1151	1	2	1
1186	0	2	1
1188	1	2	1
1227	1	2	1
1238	1	2	1
1252	6	3	1

Table 6.1. Maximum strength achieved for graphs with $n \leq 7$ vertices and XOR, Min, Maj.

Bibliography

[1] Noga Alon. “Explicit construction of exponential sized families of k-independent sets”. In: *Discrete Mathematics* 58.2 (1986), pp. 191–193.

[2] Julio Aracena. “Maximum number of fixed points in regulatory Boolean networks”. In: *Bulletin of mathematical biology* 70.5 (2008), pp. 1398–1409.

[3] Julio Aracena, Jacques Demongeot, and Eric Goles. “Fixed points and maximal independent sets in AND–OR networks”. In: *Discrete Applied Mathematics* 138.3 (2004), pp. 277–288.

[4] Julio Aracena, Adrien Richard, and Lilian Salinas. “Number of fixed points and disjoint cycles in monotone Boolean networks”. In: *SIAM journal on Discrete mathematics* 31.3 (2017), pp. 1702–1725.

[5] JA Barrau. “On the combinatory problem of Steiner”. In: *KNAW, Proceedings*. Vol. 11. 1908, pp. 1908–1909.

[6] Renée C Bryce and Charles J Colbourn. “A density-based greedy algorithm for higher strength covering arrays”. In: *Software Testing, Verification and Reliability* 19.1 (2009), pp. 37–53.

[7] Renée C Bryce and Charles J Colbourn. “The density algorithm for pairwise interaction testing”. In: *Software Testing, Verification and Reliability* 17.3 (2007), pp. 159–182.

[8] Peter Danziger et al. “Covering arrays avoiding forbidden edges”. In: *Theoretical Computer Science* 410.52 (2009), pp. 5403–5414.

[9] Jean Doyen and Alexander Rosa. “An updated bibliography and survey of Steiner systems”. In: *Annals of Discrete Mathematics*. Vol. 7. Elsevier, 1980, pp. 317–349.

[10] Maximilien Gadouleau, Adrien Richard, and Søren Riis. “Fixed points of Boolean networks, guessing graphs, and coding theory”. In: *SIAM Journal on Discrete Mathematics* 29.4 (2015), pp. 2312–2335.

[11] Niall Graham et al. “Subcube fault-tolerance in hypercubes”. In: *Information and Computation* 102.2 (1993), pp. 280–314.

[12] David G Green, Tania G Leishman, and Suzanne Sadedin. “The emergence of social consensus in Boolean networks”. In: *2007 IEEE Symposium on Artificial Life*. IEEE. 2007, pp. 402–408.

[13] Claus Kadelka, Benjamin Keilty, and Reinhard Laubenbacher. “Collectively canalizing Boolean functions”. In: *Advances in Applied Mathematics* 145 (2023), p. 102475.

[14] Ludwig Kampel and Dimitris E Simos. “A survey on the state of the art of complexity problems for covering arrays”. In: *Theoretical Computer Science* 800 (2019), pp. 107–124.

[15] Stuart A Kauffman. “Metabolic stability and epigenesis in randomly constructed genetic nets”. In: *Journal of theoretical biology* 22.3 (1969), pp. 437–467.

[16] Stuart A Kauffman. *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA, 1993.

[17] Peter Keevash. “The existence of designs”. In: *arXiv preprint arXiv:1401.3665* (2014).

[18] Daniel J Kleitman and Joel Spencer. “Families of k-independent sets”. In: *Discrete mathematics* 6.3 (1973), pp. 255–262.

[19] Boris Krupa. “On the number of experiments required to find the causal structure of complex systems”. In: *Journal of Theoretical Biology* 219.2 (2002), pp. 257–267.

[20] Jim Lawrence et al. “A survey of binary covering arrays”. In: *the electronic journal of combinatorics* (2011), P84–P84.

[21] Karen Meagher and Brett Stevens. “Covering arrays on graphs”. In: *Journal of Combinatorial Theory, Series B* 95.1 (2005), pp. 134–151.

[22] Yu V Merekin. “Upper bounds for the complexity of sequences generated by symmetric Boolean functions”. In: *Discrete applied mathematics* 114.1-3 (2001), pp. 227–231.

[23] Kari J Nurmela. “Upper bounds for covering arrays by tabu search”. In: *Discrete applied mathematics* 138.1-2 (2004), pp. 143–152.

[24] I. Osorio. “VC-dimension in Boolean networks”. MA thesis. Departamento de Ingeniería Informática. Universidad de Concepción, 2023.

[25] Ronald C Read and Robin J Wilson. *An atlas of graphs*. Vol. 21. Clarendon press Oxford, 1998.

[26] Gadiel Seroussi and Nader H Bshouty. “Vector sets for exhaustive testing of logic circuits”. In: *IEEE Transactions on Information Theory* 34.3 (1988), pp. 513–522.

[27] George Sherwood. “Effective testing of factor combinations”. In: *Proc. Third International Conference on Software Testing, Analysis and Review (STAR’94)*. 1994.

[28] Juilee Thakar. “Pillars of biology: Boolean modeling of gene-regulatory networks”. In: *Journal of Theoretical Biology* (2023), p. 111682.

[29] Alan Veliz-Cuba and Reinhard Laubenbacher. “On the computation of fixed points in Boolean networks”. In: *Journal of Applied Mathematics and Computing* 39.1 (2012), pp. 145–153.

[30] Xingyuan Wang and Suo Gao. “Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network”. In: *Information sciences* 539 (2020), pp. 195–214.