UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA CIVIL MATEMÁTICA

Relationship between interaction graph and the existence of polarized and consensus fixed points in Boolean networks

TESIS PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE CONCEPCIÓN PARA OPTAR AL GRADO DE MAGISTER DE CIENCIAS DE LA COMPUTACIÓN

Autor: Catalina Opazo

Profesor Guía: Julio Aracena

Diciembre, 2024

Contents

Capítulos **Página**

List of Figures

Chapter 1

Introduction

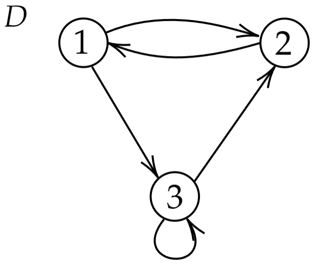
The process of binary opinion formation has been modeled using various mathematical tools, primarily non-deterministic ones (see, for example, (Grabisch & Rusinowska, 2010), (Hegselmann & Krause, 2015), (Poindron, 2021)). In these works, the relationship between network structure and the existence of stable states has been studied (Easley, Kleinberg, et al., 2010), (Li, Chen, Wang, & Zhang, 2013), (Poindron, 2021), especially consensus states (Förster, Grabisch, & Rusinowska, 2013), (Green, Leishman, & Sadedin, 2007), (Ruz & Goles, 2022), and the time it takes to reach them (Berenbrink et al., 2024). Regarding deterministic models, in (Goles, Medina, Montealegre, & Santivañez, 2022) and (Goles, Medina, & Santivañez, 2023), cellular automata are used to study consensus in majority networks. In addition to the interest in consensus stable states, polarized states, where society does not reach a consensus, have also been studied (Barrera Lemarchand, Semeshenko, Navajas, & Balenzuela, 2020). To our knowledge, there are no studies linking the interaction structure of agents with the existence of associated networks that exhibit both consensus and polarized stable states. Boolean networks have been used to model influence processes among individuals (Green et al., 2007), (Hołyst, Kacperski, & Schweitzer, 2001), (Schweitzer & Hołyst, 2000), (Plewczyński, 1998). These models are non-deterministic and define a fixed interaction function, while the network structure could change or not be relevant to the study. In this context, the stable states of the system are represented by the fixed points of the Boolean network modeling the system, which have been widely studied, as shown in (Aracena, Demongeot, & Goles, 2004), (Aracena, Richard, & Salinas, 2014), (Agur, Fraenkel, & Klein, 1988). Considering a model with Boolean networks, we assume that the interaction graph representing the influence among agents is fixed and that this influence can be positive or negative. In this article, we present sufficient and necessary conditions on the architecture of the signed digraph representing the interaction network among the agents of a system, in order to ensure the existence of local activation functions that allow the resulting global network to have polarized, consensus, or both types of fixed points at the same time. In the first section, we present the complexity of determining the existence of a polarized fixed point, as well as the necessary conditions on the interaction graph to obtain polarized fixed points. In the second chapter, conditions on the network for obtaining consensus fixed points are presented, and in the last chapter, the necessary conditions on the network structure for both types of fixed points to occur simultaneously are discussed.

Chapter 2

Definitions

2.1 Discrete math concepts

A directed graph or **digraph** D is a pair (V, A), where V is a non empty finite set of elements called vertices and A is a finite family of ordered pairs of elements from V, called arcs. Given a digraph D, we will denote $V(D)$ and $A(D)$ as its vertex and arc set respectively, furthermore, an arc (u, v) will be denoted simply as uv and if there is an arc of the form (u, u) for some $u \in V$ it will be called a loop.



$V(D) = \{1, 2, 3\}$

$A(D) = \{(1, 2), (2, 1), (1, 3), (3, 3), (3, 2)\}$

Figure 2.1: Example of a digraph with loops.

If we do not consider direction in the arcs, we can define a **simple graph** G as a pair (V, E), where V is a non-empty finite set of elements called **vertices** or **nodes**, and E is a finite set of unordered pairs of elements from V, called edges. Given a graph G, its set of vertices is denoted as $V(G)$, and its set of edges is denoted as $E(G)$. Given an edge $\{u, v\}$, we will denote it as uv or vu.

Note that a simple graph can be considered as a loopless digraph such that if there is an arc uv, there is also an arc vu, i.e. symmetric.

Given a digraph $D = (V, A)$, we will say that vertices u and v are **adjacent** or **neighbors** if there is an arc connecting them, and in the case of a simple graph, if there is an edge connecting them. Thus, given a vertex $v \in V$, we define its **out-neighborhood** $N^+(v)$ and its **in-neighborhood** $N^-(v)$ as:

$$N^+(v) = \{u \in V : vu \in A\} \qquad N^-(v) = \{u \in V : uv \in A\}.$$

In the case of a simple graph we define a vertex's v neighborhood as $N(v) = \{u \in V : uv \in E\}$.

The cardinality of a vertex's neighborhood is called its **degree**. For a digraph D and a vertex $v \in V$, we denote the in-degree and out-degree of v as $d^+(v)$ and $d^-(v)$, respectively, and they are defined as:

$$d^+(v) = |N^+(v)| \quad \text{and} \quad d^-(v) = |N^-(v)|.$$

In the case of a simple graph and a vertex v, its degree is $d(v) = |N(v)|$. The concept of degree allows us to define for a digraph $D = (V, A)$:

- Its **minimum in-degree** as $\delta^-(D) = min_{v \in V} d^-(v)$
- Its **minimum out-degree** as $\delta^+(D) = min_{v \in V} d^+(v)$
- Its **maximum in-degree** as $\Delta^-(D) = max_{v \in V} d^-(v)$
- Its **maximum out-degree** as $\Delta^+(D) = max_{v \in V} d^+(v)$

Similarly, in the case of a simple graph G, we define its minimum degree as $\delta(G) = min_{v \in V} d(v)$ and its maximum degree as $\Delta(G) = max_{v \in V} d(v)$.

Given a digraph $D = (V, A)$ and a subset of vertices $S \subseteq V$, we denote $D[S]$ as the **induced sub-digraph** by the vertices in S. Its set of vertices is S, and its set of arcs consists of those whose endpoints are both in S. Formally:

$$V(D[S]) = S \quad \text{and} \quad A(D[S]) = \{(u, v) \in A : u \in S \wedge v \in S\}.$$

We also define a **partition** of its vertices $\{V_1, V_2\}$ such that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. Moreover, we will say that a **partition is balanced** if $||V_1| - |V_2|| \leq 1$, that is, if $|V|$ is even, their sizes are equal, and if $|V|$ is odd, their sizes differ by one.

Given a digraph $D = (V, A)$, we define:

- A **directed path** of length m between vertices u and v as a sequence of vertices $u, u_1, ..., u_m = v$ such that there are no repetitions and $\forall i \in \{1, ..., m-1\}$, $u_i, u_{i+1} \in A$.

- A **directed cycle** of length m as a sequence of vertices $u_1, u_2, ..., u_m, u_{m+1} = u_1$ such that there are no repetitions except for the first and last vertex, and also $\forall i \in \{1, ..., m-1\}$, $u_i, u_{i+1} \in A$.

In the case of a simple graph $G = (V, E)$, the definitions are the same, except that we only need to have edges between each vertex in the sequence.

We also define C_n as a graph consisting of only one cycle of length n, and we define K_n as a complete graph, meaning it has all possible edges and n vertices.

2.2 Concepts of Boolean Networks

A Boolean network f is a discrete dynamical system with a finite number of agents that interact with each other. Formally, we have a system of $n \in N$ agents, where each agent $i \in \{1, ..., n\}$ is associated with a Boolean variable $x_i \in \{0, 1\}$, so $x \in \{0, 1\}^n$ is called a **configuration** or state of the system.

Each agent i updates the value of x_i at discrete time intervals according to a Boolean function $f_i : \{0, 1\}^n \to \{0, 1\}$ called the **local activation function**.

In this way, we define the **global activation function** $f : \{0, 1\}^n \to \{0, 1\}^n$ such that $f(x) = (f_1(x), f_2(x), ..., f_n(x))$. This function is called a Boolean network, the dynamics of the system it models are completely characterized by it. We say that a local activation function f_i depends on variable x_j if there exists $x \in \{0, 1\}^n$ such that:

$$f_i(x_1, ..., x_j = 0, ...x_n) \neq f_i(x_1, ..., x_j = 1, ...x_n).$$

The interaction between agents can be represented by a digraph, called the interaction graph.

Definition 1 *Given a Boolean network $f : \{0, 1\}^n \to \{0, 1\}^n$, its* ***interaction graph*** *$G(f)$ is a digraph over $[\![n]\!] := \{1, ..., n\}$ such that the arc (i, j) exists if f_j depends on x_i.*

As an example, consider the Boolean network $f : \{0, 1\}^3 \to \{0, 1\}^3$ such that $f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3, \neg x_3, \neg x_2)$. In terms of opinion formation, this network represents a system where the first agent chooses the value 1 if any other agent in the system says 1, while the second and third agents always disagree at each step. The interaction graph of the network is shown in fig. 2.2.

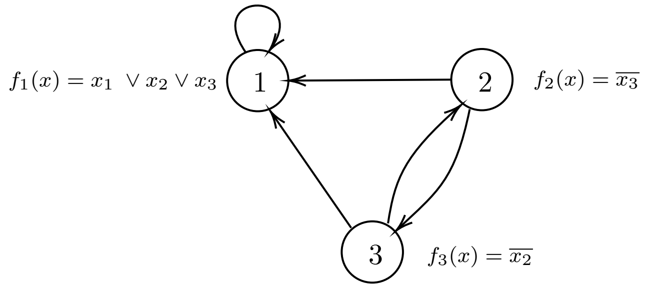


Figure 2.2: Interaction graph $G(f)$.

Definition 2 *Given* $f : \{0,1\}^n \to \{0,1\}^n$

- *A* ***fixed point*** *of* f *is a configuration* $x \in \{0,1\}^n$ *such that* $f(x) = x$.
- *A* ***polarized fixed point*** *is a fixed point such that* $|\sum_{i=1}^{n} x_i - n/2| \leq 1$
- *A* ***consensus fixed point*** *is a fixed point such that* $x_i = 1 \; \forall i \in [\![n]\!]$ *or* $x_i = 0 \; \forall i \in [\![n]\!]$.

In other words, a polarized fixed point is a configuration such that the number of components equal to 0 and the number of components equal to 1 differ by at most one. A consensus fixed point is a configuration of only ones or only zeros. It is easy to check that, in the example of fig. 2.2, the configuration 101 is a polarized fixed point.

Within the binary opinion formation process, the update rules that follow the minority or the majority of agents naturally arise. Therefore, we will define two types of minority and majority functions, which only differ in the case where the agent faces a tie in the opinions of its neighbors.

Definition 3 *Given a digraph* D *with* n *vertices, we define*

- *The associated* ***biased minority network*** *associated with* D *as* $bmin_D : \{0,1\}^n \to \{0,1\}^n$ *such that*

$$bmin_D(x)_i = H(-\sum_{j \in N^-(i)} x_j + n/2) = \begin{cases} 0 \quad if \quad \sum\limits_{j \in N^-(i)} x_j > |N^-(i)|/2 \\ \\ 1 \quad if \quad \sum\limits_{j \in N^-(i)} x_j \leq |N^-(i)|/2 \end{cases}$$

- *The associated* ***biased majority network*** *associated with* D *as* $bmay_D : \{0,1\}^n \to \{0,1\}^n$ *such that*

$$bmay_D(x)_i = H(\sum_{j \in N^-(i)} x_j - n/2) = \begin{cases} 0 \quad if \quad \sum\limits_{j \in N^-(i)} x_j < |N^-(i)|/2 \\ \\ 1 \quad if \quad \sum\limits_{j \in N^-(i)} x_j \geq |N^-(i)|/2 \end{cases}$$

Where $\forall u \in \mathbb{R}$*,* $H(u) = 1$ *if* $u \geq 0$ *and* $H(u) = 0$ *if* $u < 0$*.*

They are said to be biased because, in the case where an agent sees a tie in the opinions of its neighbors, it decides on 1. To avoid this, we define the unbiased networks, where only in case of a tie does each agent consider its own opinion from the previous time step. This implies that under certain conditions, each agent influences itself, possibly adding a loop to the interaction graph.

Definition 4 *Given a digraph* D *with* n *vertices, we define:*

- *The associated unbiased minority network or simply* ***minority network*** *associated with* D *as* $min_D : \{0,1\}^n \to \{0,1\}^n$ *such that*

$$min_D(x)_i = \begin{cases} 0 \quad if \quad \sum\limits_{j \in N^-(i) \setminus \{i\}} x_j > |N^-(i)|/2 \\ x_i \quad if \quad \sum\limits_{j \in N^-(i) \setminus \{i\}} x_j = |N^-(i)|/2 \ \vee \ N^-(i) \setminus \{i\} = \emptyset \\ 1 \quad if \quad \sum\limits_{j \in N^-(i) \setminus \{i\}} x_j < |N^-(i)|/2 \end{cases}$$

- *The associated unbiased majority network or simply* ***majority network*** *associated with* D *as* $may_D : \{0,1\}^n \to \{0,1\}^n$ *such that*

$$may_D(x)_i = \begin{cases} 0 \quad if \quad \sum\limits_{j \in N^-(i) \setminus \{i\}} x_j < |N^-(i)|/2 \\ x_i \quad if \quad \sum\limits_{j \in N^-(i) \setminus \{i\}} x_j = |N^-(i)|/2 \ \vee \ N^-(i) \setminus \{i\} = \emptyset \\ 1 \quad if \quad \sum\limits_{j \in N^-(i) \setminus \{i\}} x_j > |N^-(i)|/2 \end{cases}$$

We can think that in majority networks, agents influence each other positively, and in minority networks, agents influence each other negatively. More generally, given a local activation function $f_i(x)$, we say that:

- f_i is monotone increasing in input j if

$$\forall x \in \{0,1\}^n, x_j = 0 \implies f_i(x) \leq f_i(x + e_j)$$

- f_i is monotone decreasing in input j if

$$\forall x \in \{0,1\}^n, x_j = 0 \implies f_i(x) \geq f_i(x + e_j)$$

Where e_j is the canonical vector $x \in \{0,1\}^n$ such that $x_j = 1$ and all other components are 0.

In the first case, we say that agent j has a positive influence on agent i, and in the second case, a negative influence. Similarly, we say that the arcs connecting the vertices are positive or negative, respectively.

If all the local activation functions of a network $f : \{0,1\}^n \to \{0,1\}^n$ are either monotonically increasing or decreasing, then we say that f is **regulatory** and its interaction graph $G(f) = (V, A)$ can be assigned a sign function $\sigma_f : A \to \{-1, +1\}$ such that $\forall (i,j) \in A$, $\sigma_f(i,j) = +1$ if and only if f_j is monotone increasing in input i. We call $(G(f), \sigma_f)$ a **signed interaction graph**.

In general, we consider a signed digraph (G, σ) as a digraph that has an associated sign function $\sigma : A \to \{-1, +1\}$ assigning signs to each of its arcs. A Boolean network where all arcs of its interaction graph are positive is called **monotone**. More precisely:

Definition 5 *A Boolean network $f : \{0,1\}^n \to \{0,1\}^n$ is called **monotone** if*

$$\forall x, y \in \{0,1\}^n : x \leq y \Longrightarrow f(x) \leq f(y)$$

Where $x \leq y$ if and only if $\forall i \in [\![n]\!],\ x_i \leq y_i$.

It can be proven that a Boolean network f is monotone if and only if each local activation function f_i can be written in its conjunctive or disjunctive normal form without using any negative variables. (Anthony, 2001)

The dynamics of a Boolean network f are described by successive iterations on a configuration forming the sequence:

$$\forall t \in \mathbb{N}_0, \forall x(t) \in \{0,1\}^n, x(t+1) = f(x(t))$$

The dynamic behavior of this network can be represented by a digraph called the iteration graph $G_\Gamma(f)$, its vertex set consists of all possible configurations of $\{0,1\}^n$, and its arc set is $\{(x, f(x)) : x \in \{0,1\}^n\}$.

Definition 6 *Given a signed digraph (G, σ) and subset of vertices $I \subseteq [\![n]\!]$, we define the I-switch of (G, σ) as the signed digraph (G, σ^I) such that*

$$\sigma^I(u,v) = \begin{cases} \sigma(u,v) & if \quad \{u,v\} \subseteq I \ \vee \ \{u,v\} \cap I = \emptyset \\ -\sigma(u,v) & otherwise \end{cases}$$

Also, given a configuration $x \in \{0,1\}^n$, we define x^I such that $x_i^I = x_i$ if i^I otherwise $x_i^I = \neg x_i$.

Definition 7 *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a regulatory Boolean network and $I \subseteq [\![n]\!]$, then we can define the **dual of** f as $\overline{f}^I : \{0,1\}^n \to \{0,1\}^n$ such that $\forall x \in \{0,1\}^n$ and $\forall i \in \{1,..,n\}$:*

$$\overline{f}_i^I(x) = \begin{cases} f_i(\overline{x}^I) & if \quad i \notin I \\ \overline{f_i(\overline{x}^I)} & if \quad i \in I \end{cases}$$

Proposition 8 *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a regulatory Boolean network and $I \subseteq [\![n]\!]$, then*

i) $(G(\overline{f}^I), \sigma_{\overline{f}^I}) = (G(f), \sigma_f^I)$

ii) $G_\Gamma(f)$ *is isomorphic to* $G_\Gamma(\overline{f}^I)$

iii) $f(x) = x \iff \overline{f}^I(\overline{x}^I) = \overline{x}^I$.

Proof: Let $f : \{0,1\}^n \to \{0,1\}^n$ be a regulatory Boolean network, $I \subseteq [\![n]\!]$. To prove i) first suppose that $v \in I$, then $\overline{f}^I(\overline{x}^I) = \overline{f(x)}$ and therefore

$$\begin{aligned} uv \in A(G(f)) &\iff \exists x \in \{0,1\}^n : (f_v(x) > f_v(x+e_u)) \vee (f_v(x) < f_v(x+e_u)) \\ &\iff \exists x \in \{0,1\}^n : (\overline{f}_v^I(x) < \overline{f}_v^I(x+e_u)) \vee (\overline{f}_v^I(x) > \overline{f}_v^I(x+e_u)) \end{aligned}$$

Proving that $uv \in A(G(\overline{f}^I))$, also from here we can see that if $u \in I$, $\overline{x}_u^I \neq x_u$ and thus the sign of the arc stays the same, if $u \notin I$, $\overline{x}_u^I = x_u$ the sign of the arc changes, in both cases $\sigma_{\overline{f}^I}(u,v) = \sigma^I(u,v)$.

Now suppose that $v \notin I$, then $\overline{f}^I(\overline{x}^I) = f(x)$ and therefore

$$\begin{aligned} uv \in A(G(f)) &\iff \exists x \in \{0,1\}^n : (f_v(x) > f_v(x+e_u)) \vee (f_v(x) < f_v(x+e_u)) \\ &\iff \exists x \in \{0,1\}^n : (\overline{f}_v^I(x) > \overline{f}_v^I(x+e_u)) \vee (\overline{f}_v^I(x) < \overline{f}_v^I(x+e_u)) \end{aligned}$$

Then $uv \in A(G(\overline{f}^I))$, and similar to the previous case we can see that if $u \in I$, $\overline{x}_u^I \neq x_u$ and thus the sign of the arc changes and if $u \notin I$, $\overline{x}_u^I = x_u$ the sign of the arc stays the same, in both cases $\sigma_{\overline{f}^I}(u,v) = \sigma^I(u,v)$.

To prove $ii)$ we define $\varphi : \{0,1\}^n \to \{0,1\}^n$ such that $\forall x \in \{0,1\}^n$ $\varphi(x) = \overline{x}^I$. φ is bijective because $\forall x, \hat{x} \in \{0,1\}^n$ such that $x \neq \hat{x}$ there exists $i \in [\![n]\!]$ such that $x_i \neq \hat{x}_i$ and therefore $\varphi(x) \neq \varphi(\overline{x})$, also for every $y \in \{0,1\}^n$ there exists $x = \varphi(y)$ such that $\varphi(x) = y$.

Let $x \in \{0,1\}^n$ and let $i \in [\![n]\!]$. If $i \in I$, then

$$\overline{f}_i^I(\varphi(x)) = \overline{f}_i^I(\overline{x}^I) = \overline{f_i(x)} = \varphi_i(f(x))$$

and if $i \notin I$, then

$$\overline{f}_i^I(\varphi(x)) = \overline{f}_i^I(\overline{x}^I) = f_i(x) = \varphi_i(f(x))$$

From here we deduce that $f \circ \varphi = \varphi \circ \overline{f}^I$ and also we can see that

$$f(x) = x \iff \overline{f}^I(\varphi(x)) = \varphi(x) \iff \overline{f}^I(\overline{x}^I) = \overline{x}^I$$

proving $iii)$. $\square$

Chapter 3

Fixed Points in Boolean Networks

This chapter aims to explain what is known about fixed points in Boolean networks. Note that a network of size n has 2^n possible configurations in its dynamics, which makes it difficult to analyze using exhaustive search techniques. For this reason, the literature usually seeks to understand the dynamics of a network based on the structure of its interaction graph and its activation functions. Within this study, special attention has been given to the fixed points of the network, as they are attractors of the dynamics invariant to the update scheme used. For example, in Biology, these represent stable patterns in the expression of phenotypes. (Kauffman, 1969)

In the works of (Aracena et al., 2004), (Aracena, 2008), (Aracena et al., 2014), and (Aracena, Richard, & Salinas, 2017), the maximum number of fixed points in different families of Boolean networks is studied, such as those with local AND-OR and AND-OR-NOT activation functions, as well as the families of regulatory and monotone networks. In all these works, relationships between the structure of the interaction graph and the maximum number of fixed points are found.

Definition 9 *Given a signed digraph* (G, σ)*, its positive transversal number* $\tau^+(G)$ *is defined as the minimum number of vertices that must be removed from the digraph so that it has no positive cycles.*

Among the most relevant results, we highlight the following theorem.

Theorem 10 *(Aracena, 2008) Let* $f : \{0,1\}^n \to \{0,1\}^n$ *be a regulatory Boolean network whose signed interaction graph is* G*. The maximum number of fixed points in the network is at most* $2^{\tau^+(G)}$.

Regarding the study of the computational complexity of the problem, in (Kosub, 2008) and (Bridoux, Durbec, Perrot, & Richard, 2019), the complexity of the problem of fixed point existence in Boolean networks is studied, concluding that the problem of determining whether a given Boolean network has fixed points is NP-Complete.

In the study of monotone networks, it is known that as long as there are no constant activation functions, there is at least two consensus fixed point (Aracena, 2008). Therefore, it is interesting to ask whether there are fixed points that are not consensus. The problem is posed:

MONFIX: Given a monotone Boolean function $f : \{0,1\}^n \to \{0,1\}^n$. Does there exist a fixed point $x \in \{0,1\}^n \setminus \{0^n, 1^n\}$ of f?
Where 1^n and 0^n are configurations with only ones and zeros in their components, respectively.

In (Yang & Zhao, 2004), it is shown that MONFIX is NP-Complete. Later, we will present a sufficient condition on the minimum degree of the interaction graph of the monotone network that guarantees the existence of a non-trivial fixed point.

Given any Boolean network f, to check if there are any consensus fixed points it is enough to verify that $f(\vec{1}) = \vec{1}$ or $f(\vec{0}) = \vec{0}$. However, to determine if it has polarized fixed points is more computationally harder since there are $\binom{n}{n/2}$ polarized configurations if n is even and $\binom{n}{\lfloor n/2 \rfloor}$ if n is odd. To study the complexity of the general problem of finding polarized fixed points in a network, we present the following decision problem and a known variant of the SAT problem:

Definition 11 (POLFP) *Given a Boolean network $f : \{0,1\}^n \to \{0,1\}^n$. ¿Does f have a polarized fixed point?*

Definition 12 (HALF-SAT) *Given a formula $\phi : \{0,1\}^n \to \{0,1\}$ written in conjunctive normal form (CNF) and n is even. ¿Is there a satisfactory assignation of ϕ where half of its variables are true and the other half are false?*

It is known that this variant of SAT is also NP-Complete and the we provide the following proof.

Proposition 13 ***HALF-SAT** is NP-Complete.*

Proof: We can check in polynomial time if an assignation x with half of its variables is true and the other half is false satisfies ϕ by computing $\phi(x)$, therefore the problem is NP.

Let $\varphi : \{0,1\}^n \to \{0,1\}$ be an instance of SAT with variables $x = (x_1, x_2, ..., x_n)$, then we create a instance of HALF-SAT $\phi : \{0,1\}^{2n} \to \{0,1\}$ with variables $x' = (x_1, ..., x_n, y_1, ..., y_n)$ such that for each clause C in φ we include the clauses C and C' in ϕ, where C' obtained by replacing the literals x_i with $\neg y_i$. We can see that that $x = (x_1, ..., x_n)$ satisfies φ if and only if the assignation $x' = (x_1, .., x_n, \neg x_1, ..., \neg x_n)$ satisfies ϕ. $\square$

Proposition 14 ***POLFP** is NP-complete.*

Proof:

The problem is NP because we can verify that a configuration $x \in \{0,1\}^n$ is a polarized fixed point in polynomial time by checking if $f(x) = x$.

Let ϕ be an instance of HALF-SAT, we construct the Boolean network $f : \{0,1\}^{n+2} \to \{0,1\}^{n+2}$ such that:

- $\forall i \in [\![n]\!]$, $f_i(x) = x_i$
- $f_{n+1}(x) = \phi(x) \vee x_{n+2}$
- $f_{n+2}(x) = \neg x_{n+1}$

Let's suppose that there exists a polarized configuration $x \in \{0,1\}^n$ such that $\phi(x) = 1$. Then $\tilde{x} = (x, 1, 0)$ is a polarized fixed point of f. On the other hand, if $x \in \{0,1\}^{n+2}$ is a polarized fixed point of f, we can define $\tilde{x} \in \{0,1\}^n$ as the first n components of x. Since $x_{n+1} \neq x_{n+2}$, we see that $\tilde{x}$ is also polarized. If $x_{n+2} = 1$, then $x_{n+1} = 0$ and $f_{n+1} = 1$, but this is a contradiction with x being a fixed point, then $x_{n+2} = 0$ and $x_{n+1} = 1 = f_{n+2}(x) = \phi(x) \vee x_{n+2}$, which implies that $\tilde{x}$ satisfies ϕ. □

3.1 Polarized fixed points

Since the problem of deciding whether a Boolean network has a polarized fixed point is difficult, we study the relationship between an interaction graph and the existence of polarized fixed points in the networks associated with it. First we note that for every digraph D, we can easily construct a Boolean network associated to D that has a polarized fixed point:

Proposition 15 *Given a digraph D, there exists a Boolean network f such that $G(f) = D$ and that has at least one polarized fixed point.*

Proof: Let D be a digraph with n vertices and $x \in \{0,1\}^n$ a polarized configuration, we can define the network $f : \{0,1\}^n \to \{0,1\}^n$ such that for each $i \in [\![n]\!]$:

$$f_i(y) = \begin{cases} x_i & if \quad N^-(i) = \emptyset \\ \bigwedge\limits_{\{j \in N^-(i) : x_j = x_i\}} y_j \ \wedge \bigwedge\limits_{\{k \in N^-(i) : x_k \neq x_i\}} \neg y_k & if \quad N^-(i) \neq \emptyset \end{cases}$$

Then $f(x) = x$ and f has a polarized fixed point. $\square$

Moreover, in (Gadouleau & Richard, 2016) they exhibit a large class of unsigned digraphs that have an associated Boolean network with a unique fixed point and no other attractors such as the family of strong digraphs containing a loop or a wheel, or the family of loop-less digraphs with minimal in-degree ≥ 2 and maximal outdegree $n-1$ and the family of loopless connected symmetric digraphs with $n \geq 3$. Also they show that the families of unsigned digraphs that are a cycle or a double cycle Cl, r such that $min(l, r)$ does not divide $max(l, r)$ don't have an associated Boolean network with a unique fixed point and no other attractors.

From proposition 8 we can prove the following result.

Proposition 16 *Given a Boolean network* $f : \{0,1\}^n \to \{0,1\}^n$ *and* $x \in \{0,1\}^n$ *a fixed point of* f*, there exists* $I \subseteq [\![n]\!]$ *such that* $\overline{f}^I$ *has* x^I *as a fixed point and that keeps the dynamic behavior.*

From here, given an unsigned digraph that admits a Boolean network f with a unique fixed point as its only attractor, we can use convenient I-switch to find an associated Boolean network that has a polarized or consensus fixed point as its only attractor.

However, if we consider the type of influence each agent has over each other the problem becomes harder. To model this we include the function $\sigma : A(D) \to \{-1, +1\}$ such that the influence of agent i over agent j is positive if $\sigma(i, j) = +1$ and -1 otherwise.

First we have to consider that a signed digraph does not always have an associated Boolean network that has a fixed point. For example, if n is odd there is not a Boolean network associated to (C_n, σ) where $\sigma(e) = -1$ for all $e \in A(C_n)$ (Aracena et al., 2017). Nevertheless, we provide a sufficient and necessary condition for the architecture of the signed digraph to have a polarized fixed point.

Theorem 17 *Given a signed digraph* (D, σ)*, the following statements are equivalent:*

(i) There exists a regulatory network f *such that* $G(f) = (D, \sigma)$ *that has at least one polarized fixed point.*

(ii) D *has a balanced partition* $\{V_1, V_2\}$ *such that* $\forall i \in \{1, 2\}$*,* $\forall v \in V_i$ *it holds that* $\exists u_1 \in V_i$ *such that* $\sigma(u_1, v) = +1$ *or* $\exists u_2 \in V_{3-i}$ *such that* $\sigma(u_2, v) = -1$*.*

Proof: Let (D, σ) be a signed digraph, let's suppose that f is a Boolean network associated to D such that has x as a polarized point. We define the partition induced by x as $\{V_0, V_1\}$ where $v \in V_0 \Leftrightarrow x_v = 0$.

By contradiction let's suppose that all balanced partitions of f have a vertex that does not satisfy any condition required in (ii), in particular there must exist a vertex v in the partition induced by x such that all arcs coming within its partition set are negative and all arcs coming outside its partition set are positive. Without loss of generality, if $x_v = 1$ this means that when computing $f_v(x)$ all the negated literals take the value 0 and all positive literals also take the value 0, therefore $f(x) = 0$ which is a contradiction with x being a fixed point of f.

Now let's suppose that there exists a balanced partition $\{V_0, V_1\}$ of $V(D)$, we define the network $f : \{0,1\}^n \to \{0,1\}^n$ such that

$$f_i(x) = \begin{cases} \bigvee\limits_{j \in N^-(i)} x_j & if \quad i \in V_1 \\ \bigwedge\limits_{j \in N^-(i)} x_j & if \quad i \in V_0 \end{cases}$$

Then the configuration $x \in \{0,1\}^n$ such that $x_v = i$ if $v \in V_i$ is a fixed point of the network. $\square$

Example 18 *An example of a digraph that doesn't satisfy condition (ii) is shown in figure 3.1. We can see that in any balanced partition of this digraph there is a vertex that either receives only negative influence within its partition or positive influence from outside its partition. Therefore any Boolean network associated to it will not have a polarized fixed point.*

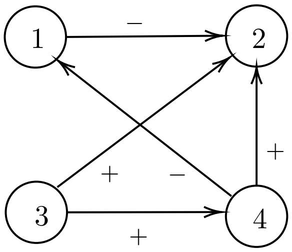


Figure 3.1: Digraph

3.1.1 Full-positive digraphs

If we only consider positive interactions between agents we call this partition valid. More precisely:

Definition 19 *Given a digraph $D = (V, A)$, we will say that a partition of its vertices $\{V_1, V_2\}$ is valid if $\delta^-(D[V_1]) \geq 1$ and $\delta^-(D[V_2]) \geq 1$.*

Corollary 20 *Given a digraph $D = (V, A)$ such that $\delta^-(D) \geq 1$. The following statements are equivalent:*

(i) There exists a monotone network f such that $G(f) = D$ and it has at least one polarized fixed point.

(ii) There exists a balanced and valid partition of V.

We can prove that the problem of deciding if there exists any Boolean network associated to a digraph that has at least one polarized fixed point is also computationally hard.

Definition 21 (BVPartition) *Given a digraph $D = (V, E)$ such that $\delta^-(D) \geq 1$. Does there exist a balanced and valid partition of V?*

Definition 22 (G1Partition) *Given a multiset S of positive integers strictly greater than 1. Can S be divided into two sets S_1 and S_2 such that $\sum_{i \in S_1} i = \sum_{i \in S_2} i$?*

Where a multiset is a set that allows repetitions.

Proposition 23 ***BVPartition*** *is NP-Complete.*

Proof: Given a partition of vertices, we can check in linear time with respect the size of G if the partition is balanced and valid, therefore **BVPartition** is NP. Let the multiset $A = \{a_1, ..., a_n\}$ be a instance of **G1Partition**, we construct a symmetric digraph $G = (V, A)$ as follows: For each a_i we create a star graph S_i with center v_i, that is, a node v_i with $a_i - 1$ hanging vertices (see Figure 3.2 for example), note that this implies that $|S_i| = a_i$. Then we add a two-cycle between v_i and v_{i+1} for each $i \in \{1, ..., n-1\}$. The resulting graph has $|V| = \sum_{i \in A} a_i$ vertices and $\delta^-(G) = 1$.

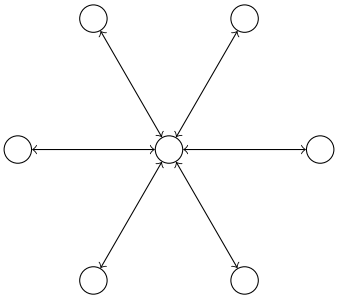


Figure 3.2: Star graph with 7 vertices and bidirectional edges.

Suppose there exists a partition $\{A_1, A_2\}$ of A such that the sum of its components is equal. We define $\{V_1, V_2\}$ of V such that $V_j = \{v \in S_i : a_i \in A_j\}$ for $j \in \{1, 2\}$. Therefore:

$$|V_1| = \sum_{i:a_i \in A_1} |S_i| = \sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i = \sum_{i:a_i \in A_2} |S_i| = |V_2|.$$

Since each star is completely contained in some set of the partition, it is clear that $\delta^-(G[V_1]) \geq 1$ and $\delta^-(G[V_2]) \geq 1$.

Now let's suppose that there exists a balanced partition $\{V_1, V_2\}$ of V that is valid, to achieve this each star must be completely contained in one of the partition sets. We define a partition $\{A_1, A_2\}$ of A such that $A_j = \{a_i \in A : v_i \in S_j\}$ for $j \in \{1, 2\}$.

$$\sum_{a_i \in A_1} a_j = \sum_{i:v_i \in V_1} |S_i| = \sum_{v \in V_1} 1 = |V_1| = |V_2| = \sum_{v \in V_2} 1 = \sum_{i:v_i \in V_2} |S_i| = \sum_{a_i \in A_2} a_i.$$

$\square$

Finding a valid and balanced partition ensures the existence of at least one associated monotone network with polarized fixed points. However, there may also be monotone networks that do not have such fixed points. For example, given a symmetric digraph if we consider the majority network associated, having just a valid and balanced partition is insufficient, see for example Figure 3.3, this digraph has a balanced valid partition $\{1, 2, 3\}, \{4, 5, 6\}$ but the associated majority network does not have polarized fixed points. To ensure the existence of polarized fixed points in majority networks associated to a digraph we need a slightly stronger condition.

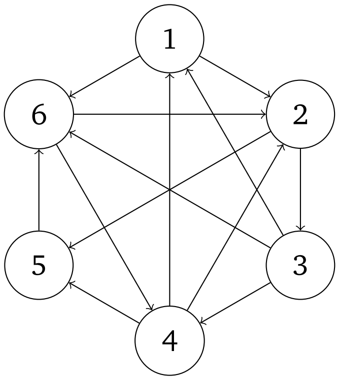


Figure 3.3: Digraph with circular arrangement of vertices.

The following is a known problem in graph theory that let us characterize when a digraph has an associated majority network with polarized fixed points.

Definition 24 *Given a digraph $G = (V, A)$ we say that a partition of its vertex $\{V_0, V_1\}$ is satisfactory* *if each vertex has at least as many in-neighbors in its own part as in the other part:*

$$\forall v \in V_i : \quad |N^-(v) \cap V_i| \geq |N^-(v) \cap V_{1-i}| \quad i \in \{0, 1\}.$$

Proposition 25 *Given a digraph $G = (V, A)$, the non biased majority network associated to G has a polarized fixed point if and only if G has a satisfactory partition that is balanced.*

Proof: Let $G = (V, A)$ be a digraph, each configuration $x \in \{0, 1\}^n$ corresponds to unique a partition $\{V_0, V_1\}$ such that $i \in V_j$ if and only if $x_i = j$ for $j = 0, 1$. We will prove that x is a fixed polarized fixed point if and only if $\{V_0, V_1\}$ is a balanced satisfactory partition.

Let's suppose that x is a polarized fixed point, then $\{V_0, V_1\}$ is balanced and for any $j \in \{0, 1\}$ each $i \in V_j$ satisfies may$(x) = j$, this happens if and only if $N(i) = \emptyset$ or $\sum_{k \in N^-(i)} x_k \geq |N^-(i)|/2$, in any case, this implies that $|N^-(i) \cap V_j| \geq |N^-(i)|/2$ and therefore $|N^-(i) \cap V_j| \geq |N^-(i) \cap V_{1-j}|$.

And now suppose that $\{V_0, V_1\}$ is a balanced satisfactory partition, x must be polarized and also for any $j \in \{0, 1\}$ lets consider $i \in V_j$, if $N^-(i) = \emptyset$ then may$_i(x) = x_i$ by definition and if $N^-(i) \neq \emptyset$ then $|N(i) \cap V_j| \geq |N(i) \cap V_{1-j}|$ which implies that may$_i(x) = \sum_{k \in N(i)} x_k = j = x_i$. □

The problem of deciding if a symmetric digraph has a satisfactory partition is NP-Complete (Bazgan, Tuza, & Vanderpooten, 2006), therefore deciding if the non biased majority network associated to a graph has a polarized fixed point is also hard.

Corollary 20 provides us with a necessary condition for the existence of a monotone Boolean network associated with a positive signed digraph such that it that has a polarized fixed point.

Corollary 26 *Given a digraph D, if D has no two vertex disjoint cycles, then there is no monotone network associated to it that has polarized fixed points.*

Example 27 *A digraph that doesn't have two vertex disjoint cycles are called intercyclic. An example of an intercyclic digraph is presented in Figure 3.4, therefore any monotone network associated to this digraph doesn't have a polarized fixed point.*

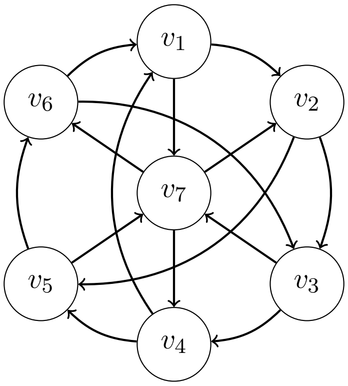


Figure 3.4: Digraph of class $\mathcal{D}_7$.

Since there is a relation between valid balanced partitions and polarized fixed points we may think that a graph with many valid balanced partitions could have an associated network with a large number of polarized fixed points. Indeed, we show a network associated to the complete graph that has an exponential number of polarized fixed points:

Proposition 28 *Let K_n be the complete graph with n vertices such that $n \mod 4 = 0$, there is a monotone Boolean network f associated to K_n that has $\binom{n/2}{n/4}$ polarized fixed points.*

Proof: Let's consider a perfect matching M of K_n. Let $i, j \in V$, such that $\{i, j\} \in E(M)$, we define for any $x \in \{0,1\}^n$:

$$f_i(x) = x_j \vee \bigwedge_{k \in [\![n]\!] \setminus \{i,j\}} x_k$$

We define A as the set of balanced partitions such that each edge of M is completely contained in some partition set, i.e.,

$$A := \{\{V_1, V_2\} : |V_1| = |V_2| \wedge \forall e \in E(M) : e \subseteq D[V_1] \vee e \subseteq D[V_2]\}$$

Let $\{V_1, V_2\} \in A$, if $x^* \in \{0,1\}^n$ such that $x_i^* = 1 \iff i \in V_1$, then x^* is a polarized fixed point of f. Indeed, let $i, j \in V$ such that $\{i, j\} \in E(M)$.

- If $v \in V_1$, then $x_j^* = 1$ and

$$f_i(x^*) = x_j^* \vee \bigwedge_{k \in [\![n]\!] \setminus \{i,j\}} x_k^* = 1 = x_i^*$$

- If $v \in V_2$, then $x_j^* = 1$, $\exists k \in V_1$ such that $\{i, k\} \in E$ and

$$f_i(x^*) = x_j^* \vee \bigwedge_{k \in [\![n]\!] \setminus \{i,j\}} x_k^* = 0 \vee 0 = 0 = x_i^*$$

That is, for each element of A, there is a polarized fixed point and $|A| = \binom{n/2}{n/4}$. $\square$

This hints that bounding the minimum degree of the interaction graph may lead us to ensure the existence of a balanced and valid partition and therefore an associated monotone network that has at least one polarized fixed point. We will prove that a sufficient condition for a symmetric digraph to have a balanced and valid partition is to have in-degree at least 2.

Given a graph $D = (V, A)$, (Schmidt, 2013) presents a procedure called chain decomposition intended to check the 2-vertex and 2-edge connectivity of G. This algorithm runs in polynomial time with respect to the size of the entry and returns a set of chains $C = \{C_1, ..., C_{|E|-|V|-1}\}$ that represent paths or cycles in G. If the graph is 2-connected, then this set represents an open ear decomposition of G, otherwise then C may not cover V or E.

To prove that any symmetric digraph of minimum in-degree 2 has a balanced and valid partition we want to compute something similar to an ear decomposition, so we made some slight modifications to the algorithm of chain decomposition previously presented so the set of chains returned cover the entire graph.

Algorithm 1 Modified Chain Decomposition

Input: $G = (V, A)$ symmetric and connected such that $\delta^-(G) \geq 2$.

```
 1: Obtain a tree $T$ from $G$ using DFS starting from a vertex that is part of a cycle.
 2: $Chains \leftarrow [\ ]$
 3: Mark all vertices as unvisited.
 4: for $v \in V$ in DFS order do
 5:   Mark $v$ as visited
 6:   for $u \in N(v)$ do
 7:     if $u$ has not been visited and is not a child of $v$ in $T$ then
 8:       Mark $u$ as visited
 9:       $C \leftarrow [v, u]$
10:       $w \leftarrow$ parent of $u$ in $T$
11:       while $w$ has not been visited do
12:         Mark $w$ as visited
13:         Add $w$ to the end of $C$
14:         $w \leftarrow$ parent of $w$ in $T$
15:       end while
16:       Add $w$ to the end of $C$
17:       if $w$ is not the root of $T$ and $w$ is not part of any chain then
18:         $w \leftarrow$ parent of $w$ in $T$
19:         while $w$ is not part of any chain do
20:           Add $w$ to the end of $C$
21:           $w \leftarrow$ parent of $w$ in $T$
22:         end while
23:         Add $w$ to the end of $C$
24:       end if
25:       $Chains \leftarrow Chains \cup \{C\}$
26:     end if
27:   end for
28: end for
29: return $Chains$
```

The only modification made to the original algorithm is in lines 17 to 24, here we keep adding vertices to the chain until it reaches a vertex that is part of another chain. This way every vertex is part of one chain at least, but now a chain can be a cycled attached to a path.

The algorithm returns a set of chains $C = \{C^1, C^2, ..., C^m\}$, where each chain $C^i : c^i_1, ..., c^i_{n_i}$ is a list of vertices of G. C_1 is a cycle and the remaining chains share one or two vertices with another chain that has a smaller index, we call these roots r_1 and r_2. Excluding the roots, the chains are vertex-disjoint from each other. More specifically, for $i \neq 1$:

- C^i is a path if $c^i_1 \neq c^i_{n_i}$ and there are no other repetitions among the elements. In this case, the roots are $r_1 = c^i_1$ and $r_2 = c^i_{n_i}$.
- C^i is a cycle if $c^i_1 = c^i_{n_i}$, here the only root is this vertex and $r_1 = r_2 = c^i_1$.
- C^i is a cycle connected to a path if $c^i_1 \neq c^i_{n_i}$ and the first vertex repeats before the last element. In this case, $r_1 = r_2 = c^i_{n_i}$.

Definition 29 *Let $C^i \in C$, the vertices that are not the roots we call internal and this set is denoted as* $int(C^i) := C^i \setminus \{r_1, r_2\}$.

The length of the chain is the number of internal vertices and we denote it as $len(C^i) = |int(C^i)|$.

Now we present the following procedure that uses the chain decomposition obtained by algorithm 1 to find a valid and balanced partition when the input is a symmetric digraph of minimum in-degree at least 2.

Algorithm 2 Valid Balanced Partition

```
Input: $G = (V, A)$ symmetric and connected such that $\delta^-(G) \geq 2$.
Obtain chain decomposition $C^1, ..., C^m$ of $G$.
$V_1 \leftarrow [\,]$, $V_2 \leftarrow [\,]$
if $len(C^1) \geq 4$ then
    for $i \in \{1, ..., \lfloor \frac{len(C^1)}{2} \rfloor\}$ do
        $V_1 \leftarrow V_1 \cup \{c_i^1\}$
    end for
    for $i \in \{\lfloor \frac{len(C^1)}{2} \rfloor\} + 1, ..., len(C^1)\}$ do
        $V_2 \leftarrow V_2 \cup \{c_i^1\}$
    end for
else
    for $i \in \{1, 2, 3\}$ do
        if $c_1^i$ is root of another chain then
            $r \leftarrow i$
            Break
        end if
    end for
    $V_1 \leftarrow V_1 \cup \{r\}$
    $V_2 \leftarrow V_2 \cup (int(C^1) \setminus \{r\})$
end if
for $i \in \{2, ..., m\}$ do
    if $r_1^i \in V_1$ o $r_2^i \in V_1$ then
        $V_1 \leftarrow V_1 \cup int(C_i)$
    else
        $V_2 \leftarrow V_2 \cup int(C_i)$
    end if
end for
Balance $\{V_1, V_2\}$
return $\{V_1, V_2\}$
```

Algorithm 3 Balance $\{V_1, V_2\}$

```
Input: $G = (V, A)$ symmetric and connected and $\{V_1, V_2\}$ a valid partition of V.
if $|V_1| - |V_2| > 1$ then
    $V_1' \leftarrow V_2$
    $V_2 \leftarrow V_1$
    $V_1 \leftarrow V_1'$
end if
while $||V_2| - |V_1|| > 1$ do
    $B \leftarrow \{v \in V_2 : \exists w \in V_1 : w \in N(v)\}$
    for $v \in B$ do
        $flag = 0$
        for $w \in N(v) \cap V_2$ do
            if $|N(w) \cap V_2| == 1$ then
                $flag = 1$
            end if
        end for
        if $flag == 0$ then
            $V_1 \leftarrow V_1 \cup \{w\}$, $V_2 \leftarrow V_2 \setminus \{w\}$
            Break
        end if
    end for
end while
return $\{V_1, V_2\}$
```

Theorem 30 *Given a symmetric connected digraph $G = (V, A)$ such that $|V| \geq 4$ and $\delta^-(G) \geq 2$, there exists a valid balanced partition of V.*

Proof: Let $G = (V, A)$ be a symmetric digraph such that $|V| \geq 4$ and $\delta^-(G) \geq 2$, we will prove that algorithm 2 returns a valid balanced partition of G in polynomial time.

First we get a chain decomposition of G, in polynomial time, using algorithm 1. Steps 3 to 19 divide the first chain in half between the two partition sets declared in step 2. Since C^1 is a cycle, if it has more than 3 vertices the partition obtained by putting the first half in V_1 and the rest in V_2 is valid and if it has exactly 3 vertices, one of them must be the root of another chain, we place this vertex in V_1 and the rest in V_2. Steps 20 to 26, place each chain by index order in any partition set where they have a root in, this way $\{V_1, V_2\}$ form a valid but not balanced partition. Finally step 27 uses algorithm 3 to balance it.

For any $i \in \{1, 2\}$ we define the action of switching a vertex from V_i to V_{3-i} as redefining $V_i =$

$V_i \setminus \{v\}$ and $V_{3-i} = V_{3-i} \cup \{v\}$. It is clear that switching a vertex v keeps the partition valid if and only if v has at least one neighbor in V_{3-i} and every neighbor of v in V_i has another neighbor in this set different than v. Algorithm 3 receives a valid partition of G and rename the partition sets such that $|V_2| - |V_1| > 1$, then in steps $6 - 20$ we define B as the set of vertices of V_2 that have at least one neighbor in V_1. We will prove that we can always switch one or two vertices from B keeping $\{V_1, V_2\}$ valid until is balanced.

Since all internal vertices of each chain are disjoint, in every step we could choose $w \in B$ that belongs to a chain with the smallest index number C^k. If either $|V_2 \cap int(C^k)| > 2$ or $|V_2 \cap int(C^k)| = 2$ and one of $int(C^k)$ is root of another chain contained in V_2, it is clear that switching w keeps $\{V_1, V_2\}$ valid. If $|V_2 \cap int(C^k)| = 2$ and none of $int(C^k)$ is root of another chain contained in V_2 then we can switch both vertices from $V_2 \cap int(C^k)$ and keep $\{V_1, V_2\}$ valid.

The only problem that may arise is when $|V_2 \cap int(C^k)| = 2$, none of $int(C^k)$ is root of another chain contained in V_2 and we only need to switch one more vertex from V_2 to balance the partition. In this case we could choose another vertex of B that belongs to the next chain with the smallest index that has more than two internal vertices in V_2. If there is not a chain like that, then every chain remaining in V_2 has only two internal vertex in this set and its roots in V_1 (i.e. $G[V_2]$ is a perfect matching), in this case we could return to the previous step and switch back from V_1 to V_2 the last vertex we chose from B and then switch one of these chains from V_2 to V_1 to balance $\{V_1, V_2\}$ (see figure 3.5). This algorithm runs in polynomial time too. ☐

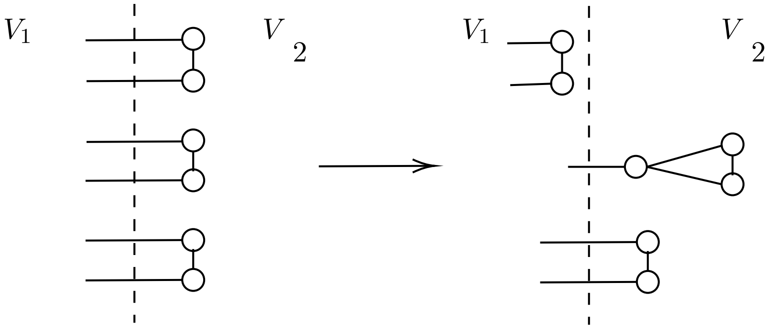


Figure 3.5: Chain switch

These algorithms were implemented and tested in Python A.3. Also, in (Maurer, 1979) there is a coloring algorithm that returns a valid balanced partition of the vertices for n even, it starts with a balanced partition and they make it valid by switching vertices. The difference with this algorithm lies in the use of chain decomposition and also it doesn't require n to be even.

3.1.2 Full-negative digraphs

It is known that if a regulatory network has at least one fixed point then its interaction graph must have at least one positive cycle (Aracena, 2008), therefore if we only consider negative interactions between agents we may not have any fixed points. Nevertheless the characterization in theorem 17 in the case of only negative signs is enough to ensure the existence of a positive cycle.

A particular example of a network where all agents have negative influence between each other is a minority network, to ensure the existence of polarized fixed points we need more than the condition presented in 17. To do this, we present a known problem in graph theory:

Definition 31 *A digraph $D = (V, A)$ has a majority coloring if it can be colored with two colors such that each vertex has the same color as at most half of its out neighbors. More precisely, $\phi : V \to \{0, 1\}$ is a majority coloring if*

$$\forall v \in V, |\{u \in N^+(v) : \phi(u) = \phi(v)\}| \leq |N^+(v)|/2$$

Additionally, we will say that a digraph has a balanced majority coloring if the partition that this coloring induces is balanced.

Proposition 32 *Given a digraph $D = (V, A)$, the following statements are equivalent:*

1. *The associated minority network has at least one polarized fixed point.*
2. *D^r has a balanced majority coloring $\phi : V \to \{0, 1\}$.*

Where $D^r = (V, A^r)$ is the reverse digraph of D, that is, $\forall u, v \in V, \ \ (u, v) \in A^r \iff (v, u) \in A$.

Proof: Let $D = (V, A)$ be a digraph with n vertices. If the associated minority network min has a polarized fixed point x, then we define $\phi : V \to \{0, 1\}$ such that $\phi(i) = x_i$.

We will prove that ϕ is a balanced coloring of D^r. Let $v \in V$, we denote $N_D^-(v)$ as the incoming neighborhood of v in D and $N_{D^r}^+(v)$ as the outgoing neighborhood of v in D^r. It is clear that $N_{D^r}^+(v) = N_D^-(v)$.

$$\begin{aligned}
\min_v(x) = x_v &\implies |\{u \in N_D^-(v) : x_v = x_u\}| \leq |N_D^-(v)|/2 \\
&\implies |\{u \in N_{D^r}^+(v) : x_v = x_u\}| \leq |N_{D^r}^+(v)|/2 \\
&\implies |\{u \in N_{D^r}^+(v) : \phi(v) = \phi(u)\}| \leq |N_{D^r}^+(v)|/2
\end{aligned}$$

Now, suppose that D^r has a balanced majority coloring $\phi : V \to \{0,1\}$, then we will prove that $x \in \{0,1\}^n$ such that $\forall i \in V$, $x_i = \phi(i)$ is a polarized fixed point of the associated unbiased minority network of D, $\min_D$.

Let $v \in V$,

$$\begin{aligned} |\{u \in N^+_{D^r}(v) : \phi(v) = \phi(u)\}| \leq |N^+_{D^r}(v)|/2 &\Longrightarrow |\{u \in N^+_{D^r}(v) : x_u = x_v\}| \leq |N^+_{D^r}(u)|/2 \\ &\Longrightarrow |\{u \in N^-_{D}(v) : x_u = x_v\}| \leq |N^-_{D}(v)|/2 \\ &\Longrightarrow \min_v(x) = x_v \end{aligned}$$

□

We can prove that given a digraph D, to decide if it has a balanced majority coloring is NP-Complete and therefore to know if it has a polarized fixed point is also difficult. If we consider a symmetric digraph there are some families of graphs that ensure the existence of polarized fixed points in the associated minority networks.

Lemma 33 *Let G be a k-regular bipartite graph with $k \geq 1$. If the partition is $\{U, W\}$, then $|U| = |W|$.*

Proof: Since G is bipartite, all the edges of the graph can be counted by summing the degrees of a single partition as follows:

$$\begin{aligned} \sum_{v \in U} d(v) = \sum_{v \in W} d(v) = |E| \\ \Longrightarrow \sum_{v \in U} k = \sum_{v \in W} k \\ \Longrightarrow k|U| = k|W| \\ \Longrightarrow |U| = |W| \end{aligned}$$

□

We call balanced bipartite graphs those that have a partition $\{U, W\}$ such that $|U| = |W|$.

Lemma 34 *Let G be a balanced bipartite connected graph, then minority networks associated to it have at least one polarized fixed point.*

Proof: Let $G = (V, E)$ be a balanced bipartite graph with partition $\{U, U^c\}$. We define $x \in \{0,1\}^n$ such that $x_i = 1$ if and only if $i \in U$, since all the neighbors of each vertex are in the opposite set each agent keeps its opinion and x is a polarized fixed point of both the minority network and the biased minority network associated to G. $\square$

Using both lemmas, it follows that the associated minority networks with and without bias of a regular bipartite graph have at least one polarized fixed point.

Theorem 35 *Let G be a k-regular bipartite graph, then the associated unbiased minority network and the biased minority network have at least one polarized fixed point.*

Also, we can ensure the existence of polarized fixed points by bounding the minimum and maximum degree.

Proposition 36 *Given a digraph D with n vertices such that n is even and $\Delta^-(D) \leq 2$, the associated minority network has at least one polarized fixed point.*

Proof: Let D digraph such that $\Delta^-(D) \leq 2$, then it must be composed of its connected components that are isolated vertices, cycles and paths.

We can find a polarized fixed point in this network by enumerating the vertices from 1 to n such that connected component has consecutive vertices, then the configuration $x \in \{0,1\}^n$ such that $x_i = 1$ if and only if i is even is a fixed point. To prove the latter, let C be a connected component of D; If C is an isolated vertex, then by definition it keeps its opinion; If C is a path or an even length cycle then each vertex sees only neighbors with the opposing opinion and keeps its opinion and finally if C is an odd length cycle then only the first and last enumerated vertices see a tie between the opinion of its neighbors and keeps it opinion, the rest only see opposing opinions and therefore keeps its opinion. $\square$

Proposition 37 *Given a digraph D with n vertices and loops in all its vertices such that n is even and $\delta^-(D) \geq n-1$, the associated unbiased minority network min$_D$ has $\binom{n}{n/2}$ polarized fixed points.*

Proof: Let $D = (V, A)$ be a digraph on n vertices and $x \in \{0,1\}^n$ be any polarized configuration and consider the associated minority function. Let $v \in V$, if $d^-(v) = n-1$, the vertex is connected to all the vertices of the graph and therefore it sees exactly $\frac{1}{2} < \frac{|N(v)|}{2}$ vertices with opposing opinion and keeps its opinion. If $d(v) = n-2$ then its connected to all vertices of the graph but one that we call k, if $x_v = x_k$ then v sees $\frac{n}{2} < \frac{n}{2} - 1 = \frac{|N(v)|}{2}$ and if $x_v \neq x_k$ then v sees $\frac{n}{2} - 1 = \frac{|N(v)|}{2}$, either way keeps its opinion. Therefore x is polarized fixed point of the minority function associated to D

Since the maximum number of different polarized configurations with n elements is $\binom{n}{n/2}$, we conclude that this is the size of the set of polarized fixed points of $\min_D$. $\square$

We developed a Python program to study the case of minority networks, this program tests every possible polarized configuration to see if its a fixed point. See appendix A.1A.2.

3.2 Consensus Fixed Points

Unlike polarized fixed points, there are only two consensus configurations: $\vec{0}$ and $\vec{1}$, so determining whether a network has any of these fixed points is straightforward.

First, we observe that given a digraph D, we can always obtain a network that has at least one polarized fixed point.

Proposition 38 *Given a digraph D, there exists a Boolean network f such that $G(f) = D$ and f has at least one consensus fixed point.*

Proof: Given a digraph D over $[\![n]\!]$, we can define the network $f : \{0,1\}^n \to \{0,1\}^n$ such that for each $i \in [\![n]\!]$:

$$f_i(x) = \begin{cases} 1 & if \quad N^-(i) = \emptyset \\ \\ \bigvee\limits_{\{j \in N^-(i)\}} x_j & if \quad N^-(i) \neq \emptyset \end{cases}$$

It is easy to verify that $f(\vec{1}) = \vec{1}$, so it has a consensus fixed point. $\square$

However, if we have a digraph with signs on its edges, such a network does not always exist. In (Aracena, 2008), the following lemma is proven.

Lemma 39 *Given a Boolean regulatory network f with $G(f) = (V = [\![n]\!], A)$, if x is a fixed point, then every vertex $v \in V$ such that $N^-(v) \neq \emptyset$ receives at least one non-frustrated arc. That is:*

$$\exists w \in N^-(v) : x_v = x_w \Longleftrightarrow \sigma(w, v) = +1$$

Using this, we can infer the following condition on the interaction graph that ensures the existence of a network with at least one consensus fixed point:

Corollary 40 *Given a signed digraph (D, σ), there exists a Boolean regulatory network f associated to it that has at least one consensus fixed point if and only if each vertex that does not have zero in-degree receives at least one positive arc.*

Proof: Let (D, σ) be a signed digraph over $[\![n]\!]$. The first implication follows directly from Lemma 39. To prove the other implication, suppose that each vertex that has a neighbor receives a positive arc. We can define the networks $f, g : \{0,1\}^n \to \{0,1\}^n$ such that

$$f_i(x) = \bigvee_{j \in I_i^+} x_j \vee \bigvee_{j \in I_i^-} \overline{x_j} \qquad \text{if } N^-(i) \neq \emptyset$$

and $f_i(x) = 1$ otherwise. Where $I_i^+ = \{j \in N^-(i) : \sigma(j,i) = +1\}$ and $I_i^- = N(i) \setminus I_i^+$. Additionally,

$$g_i(x) = \bigwedge_{j \in I_i^+} x_j \vee \bigwedge_{j \in I_i^-} \overline{x_j} \qquad \text{if } N^-(i) \neq \emptyset$$

and $g_i(x) = 0$ otherwise. It is easy to verify that $f(\vec{1}) = \vec{1}$ and $g(\vec{0}) = \vec{0}$. $\square$

We see that if there are vertices with zero in-degree, we cannot have more than one fixed point. However, to ensure the existence of a network that has both consensus fixed points $\vec{0}$ and $\vec{1}$ we need more requirements.

Proposition 41 *Given a signed digraph (D, σ), there exists a Boolean regulatory network f associated to it such that it has both $\vec{0}$ and $\vec{1}$ as fixed points if and only if $\delta^-(D) \geq 1$ and every vertex that receives a negative arc also receives at least two positive ones.*

Proof: Let (D, σ) be a signed digraph. If there exists a network f such that $G(f) = (D, \sigma)$ and it has the fixed points $\vec{0}$ and $\vec{1}$, then by Lemma 39, every vertex must receive at least one positive arc.

Suppose, by contradiction, that there is a vertex v that receives negative influence from other vertices, and $\exists! w \in N^-(v)$ such that $\sigma(w, v) = +1$. Let ϕ be an irreducible disjunctive normal form (DNF) representation of f_v. Since $f_v(\vec{1}) = 1$, there must be a clause that only contains x_w; otherwise, $f_v(\vec{1}) = 0$. Note that any other clause C_i containing x_w can be removed since $C_i = 1 \implies x_w = 1$. As ϕ is irreducible, the rest of the clauses only contain negative literals, implying that $f_v(\vec{0}) = 1$, which contradicts the fact that $\vec{0}$ is a fixed point.

On the other hand, suppose that each vertex $v \in V(D)$ that receives a negative arc also receives at least two positive arcs from v_1 and v_2. We define $f_v : \{0,1\}^n \to \{0,1\}$ such that $f_v(x) = x_{v_1} \vee \bigwedge_{i \in N^-(v) \setminus \{v_1\}} \overline{x_i}$. We can verify that $f_v(\vec{0}) = 0$ and that $f_v(\vec{1}) = 1$. Hence, the global network f has $\vec{0}$ and $\vec{1}$ as fixed points. $\square$

This last result implies that in the particular case where all arcs of a signed digraph are positive, it is always possible to find a network associated to it with at least one consensus fixed point, and if the minimum in-degree is at least 1 it is always possible to find a network associated to it that has both consensus fixed points. Moreover, if all arcs are negative, there are not associated networks with consensus fixed points.

Note also that both conditions can be verified in linear time with respect to the size of the given digraph.

3.3 Polarized and Consensus Fixed Points

Given an interaction digraph, finding a network associated to it that has both types of fixed points, consensus and polarized, at the same time is more complicated than it seems.

A sufficient condition when the digraph is symmetric is that $\delta^{-}(D) \geq 2$, as we can construct a monotone network that has both types of fixed points, but it is not a necessary condition.

Example 42 *For example, the network* $f(x) = (x_2, x_1 \wedge x_5, x_1, x_1 \vee x_5, x_4 \vee x_6, x_6)$ *has* $(0,0,0,1,1,1)$ *as a fixed point, but its interaction graph has a minimum degree of 1, as shown in figure 3.6.*

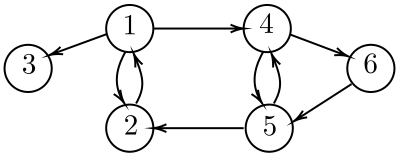


Figure 3.6: Interaction Graph

A necessary condition for the existence of a network with more than one fixed point is that the graph is not acyclic (Robert, 2012). However, this is not a sufficient condition, since if there are vertices with in-degree 1, for a consensus fixed point to exist, the interaction can only be positive. But in this case, polarized fixed points do not always exist. For example, if D is a directed cycle of length n, there is no associated network that has both a polarized and consensus fixed point, see Theorem 17.

We can demonstrate that the general problem of finding an associated Boolean network that has both types of polarized and consensus fixed points is NP-Complete. To do this, we define the following decision problems:

Definition 43 (POLCONFP) *Given a digraph D, decide whether there exists a Boolean network f associated to it such that it has at least a consensus fixed point and a polarized fixed point.*

Definition 44 (PARTITION) *Given a multiset* S *of positive integers, can* S *be divided into two sets* S_1 *and* S_2 *such that* $\sum_{i \in S_1} i = \sum_{i \in S_2} i$*?*

Proposition 45 ***POLCONFP*** *is NP-Complete.*

Proof: Given a digraph $D = (V, A)$, we consider the certificate $\{f, x\}$, which is a Boolean network associated with D and x is its polarized fixed point. To check if it has a consensus fixed point, we need to verify that $f(\vec{0}) = \vec{0}$ or $f(\vec{1}) = \vec{1}$ and that $f(x) = x$.

Moreover, **PARTITION** $\leq_P$ **POLCONFP**. Given a multiset $S = \{a_1, ..., a_m\}$, we define the digraph D as the union of m vertex disjoint directed cycles. For each component a_i, we add a directed cycle C_i with $2a_i$ vertices to D.

Suppose (S_1, S_2) is a partition of S that satisfies $\sum_{a_i \in S_1} a_i = \sum_{a_i \in S_2} a_i$. Then $V_1 := \{V(C_i) \subseteq V : a_i \in S_1\}$ and $V_2 := V \setminus V_1$ is a valid partition of D because each component C_i is entirely contained in some partition set, and it is also balanced because

$$|V_1| = \sum_{i:a_i \in S_1} |V(C_i)| = \sum_{a_i \in S_1} 2a_i = \sum_{a_i \in S_2} 2a_i = \sum_{i:a_i \in S_2} |V(C_i)| = |V_2|$$

On the other hand, suppose there exists a Boolean network f such that $G(f) = D$ and it has both a polarized and consensus fixed point. First, note that for all $v \in V$, $d^-(v) = 1$, and since the network has a consensus fixed point, all interactions must be positive (see Corollary 40). Since all interactions are positive, f has a polarized fixed point if there exists a valid and balanced partition of D (V_1, V_2). Then each component C_i must be entirely contained in one partition set, so we can define $S_1 := \{s_i \in S : C_i \in V_1\}$ and $S_2 := S \setminus S_1$.

Since (V_1, V_2) is balanced and $|V|$ is even, $|V_1| = |V_2|$, and

$$\sum_{a_i \in S_1} a_i = \sum_{i:a_i \in S_1} |V(C_i)/2| = |V_1|/2 = |V_2|/2 = \sum_{i:a_i \in S_2} |V(C_i)/2| = \sum_{a_i \in S_2} a_i$$

$\square$

3.4 Conclusion

We have shown that the existence of polarized and consensus fixed points in Boolean networks impose specific structural properties over the interaction graph. In particular, we provided necessary and sufficient conditions for a digraph to have an associated network with these fixed points, with a special focus on networks where the signs of the interaction graph are predetermined.

Among the most important result we identified that a signed digraph admits a network with a polarized fixed point if it has a balanced partition where every vertex satisfies certain conditions formalized in theorem 17 and if all the signs are positive then a sufficient condition for a symmetric digraph is to have minimum in-degree at least 2. In the particular case of majority (full-positive digraph) and minority (full-negative digraph) we need additional conditions, these were associated to known NP-Complete problems in graph theory. In proposition 41 we provided clear criteria linking the presence of positive arcs and negative arcs to the existence of both consensus fixed points in associated networks.

These theoretical advances are accompanied by algorithm contributions that allow us to find polarized fixed points in Boolean networks under specific graph constrains. We developed some Python programs to implement and validate the algorithms proposed in this work.

Concerning future work, we conjecture that a sufficient condition for a non symmetric digraph to have an associated network with a polarized fixed point is to have minimum in-degree at least 3. In the unsigned case we were able to create an associated network with a polarized or consensus fixed point as its only attractor when possible, we could study under which conditions this can be done in the signed case too. Finally we could study how conditions from 17 change for some other families of Boolean networks that could appear in the modelling of opinion forming systems, such as homogeneous Boolean networks.

References

Agur, Z., Fraenkel, A. S., & Klein, S. T. (1988). The number of fixed points of the majority rule. *Discrete Mathematics*, *70*(3), 295–302.

Anthony, M. (2001). *Discrete mathematics of neural networks: selected topics*. SIAM.

Aracena, J. (2008). Maximum number of fixed points in regulatory boolean networks. *Bulletin of mathematical biology*, *70*, 1398–1409.

Aracena, J., Demongeot, J., & Goles, E. (2004). Fixed points and maximal independent sets in and–or networks. *Discrete Applied Mathematics*, *138*(3), 277–288.

Aracena, J., Richard, A., & Salinas, L. (2014). Maximum number of fixed points in and–or–not networks. *Journal of Computer and System Sciences*, *80*(7), 1175–1190.

Aracena, J., Richard, A., & Salinas, L. (2017). Number of fixed points and disjoint cycles in monotone boolean networks. *SIAM journal on Discrete mathematics*, *31*(3), 1702–1725.

Barrera Lemarchand, F., Semeshenko, V., Navajas, J., & Balenzuela, P. (2020). Polarizing crowds: Consensus and bipolarization in a persuasive arguments model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *30*(6).

Bazgan, C., Tuza, Z., & Vanderpooten, D. (2006). The satisfactory partition problem. *Discrete applied mathematics*, *154*(8), 1236–1245.

Berenbrink, P., Hoefer, M., Kaaser, D., Lenzner, P., Rau, M., & Schmand, D. (2024). Asynchronous opinion dynamics in social networks. *Distributed Computing*, 1–18.

Bridoux, F., Durbec, N., Perrot, K., & Richard, A. (2019). Complexity of maximum fixed point problem in boolean networks. In *Conference on computability in europe* (pp. 132–143).

Easley, D., Kleinberg, J., et al. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world* (Vol. 1). Cambridge university press Cambridge.

Förster, M., Grabisch, M., & Rusinowska, A. (2013). Anonymous social influence. *Games and Economic Behavior*, *82*, 621–635.

Gadouleau, M., & Richard, A. (2016). Simple dynamics on graphs. *Theoretical Computer Science*, *628*, 62–77.

Goles, E., Medina, P., Montealegre, P., & Santivañez, J. (2022). Majority networks and consensus dynamics. *Chaos, Solitons & Fractals*, *164*, 112697.

Goles, E., Medina, P., & Santivañez, J. (2023). Majority networks and local consensus algorithm. *Scientific Reports*, *13*(1), 1858.

Grabisch, M., & Rusinowska, A. (2010). A model of influence in a social network. *Theory and Decision*, *69*, 69–96.

Green, D. G., Leishman, T. G., & Sadedin, S. (2007). The emergence of social consensus in boolean networks. In *2007 ieee symposium on artificial life* (pp. 402–408).

Hegselmann, R., & Krause, U. (2015). Opinion dynamics under the influence of radical groups, charismatic leaders, and other constant signals: A simple unifying model. *Networks and Het-*

erogeneous Media, *10*(3), 477–509.

Hołyst, J. A., Kacperski, K., & Schweitzer, F. (2001). Social impact models of opinion dynamics. *Annual Reviews Of Computational PhysicsIX*, 253–273.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, *22*(3), 437–467.

Kosub, S. (2008). Dichotomy results for fixed-point existence problems for boolean dynamical systems. *Mathematics in Computer Science*, *1*(3), 487–505.

Li, Y., Chen, W., Wang, Y., & Zhang, Z.-L. (2013). Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the sixth acm international conference on web search and data mining* (pp. 657–666).

Maurer, S. B. (1979). Vertex colorings without isolates. *Journal of Combinatorial Theory, Series B*, *27*(3), 294–319.

Plewczyński, D. (1998). Landau theory of social clustering. *Physica A: Statistical Mechanics and its Applications*, *261*(3-4), 608–617.

Poindron, A. (2021). A general model of binary opinions updating. *Mathematical Social Sciences*, *109*, 52–76.

Robert, F. (2012). *Discrete iterations: a metric study* (Vol. 6). Springer Science & Business Media.

Ruz, G. A., & Goles, E. (2022). Characterizing consensus in threshold boolean networks. In *2022 international joint conference on neural networks (ijcnn)* (pp. 1–8).

Schmidt, J. M. (2013). A simple test on 2-vertex-and 2-edge-connectivity. *Information Processing Letters*, *113*(7), 241–244.

Schweitzer, F., & Hołyst, J. A. (2000). Modelling collective opinion formation by means of active brownian particles. *The European Physical Journal B-Condensed Matter and Complex Systems*, *15*, 723–732.

Yang, K., & Zhao, Q. (2004). The balance problem of min–max systems is co-np hard. *Systems & control letters*, *53*(3-4), 303–310.

Appendix **A**

Algorithms

A.1 Program that finds polarized fixed points in a non-biased minority network

This program receives a symmetric digraph G and function *exPF(G)* returns a polarized fixed point of the network in case that it exists and 0 otherwise. To do this we test every possible polarized configuration. It was used at the beginning of the investigation to explore examples and look for patterns for section 3.1.2.

```python
#Non biased minority function

def minsinb(n,nv,x):
    sum=0
    for i in nv:
        sum=sum+x[i-1]
    if sum < len(nv)/2:
        return(1)
    elif sum > len(nv)/2:
        return(0)
    else:
        return(x[n-1])

#Decides if x is a fixed point
def ispf(G,x):
    for i in G:
        if minsinb(i,G[i],x) != x[i-1]:
```

```
                return(-1)
    return(1)

#Generates all possible polarized configurations of size n
from itertools import combinations
def genP(n):
    lst=[]
    for i in range(0,n):
        if i <n/2:
            lst.append(0)
        else:
            lst.append(1)
    return(lst)
def polComb(n):
    lst=genP(n)
    for comb in combinations(range(n), int(n/2)):
        result = [0]*n
        for i in comb:
            result[i] = 1
        yield result

#Search for the first polarized configuration that is a fixed point
def exPF(G):
    n=len(G)
    for comb in polComb(n):
        if ispf(G,comb) == 1:
            return(comb)
    return(0)
```

A.2 Program that finds polarized fixed points in a biased minority network

```
#Biased minority function
def umb(nv,x):
    sum=0
    for i in nv:
        sum=sum+x[i-1]
```

```
    if sum <= len(nv)/2:
        return(1)
    else:
        return(0)

#Decides if x is a fixed point
def ispf(G,x):
    for i in G:
        if umb(G[i],x) != x[i-1]:
            return(-1)
    return(1)

def genP(n):
    lst=[]
    for i in range(0,n):
        if i <n/2:
            lst.append(0)
        else:
            lst.append(1)
    return(lst)

#Generates all possible polarized configurations of size n
from itertools import combinations
def polComb(n):
    lst=genP(n)
    for comb in combinations(range(n), int(n/2)):
        result = [0]*n
        for i in comb:
            result[i] = 1
        yield result

#Search for the first polarized configuration that is a fixed point
def exPF(G):
    n=len(G)
    for comb in polComb(n):
        if ispf(G,comb) == 1:
            return(comb)
    return(0)
```

A.3 Program to find a biased balanced partition

```

#DFS

def DFSrec(G,v,visited,dfstree,dfsindex):
    visited[v-1]=1
    dfsindex.append(v)
    for u in G[v]:
        if visited[u-1]==0:
            dfstree[u]=v
            DFSrec(G,u,visited,dfstree,dfsindex)

def DFS(G,v):
    dfstree={}
    visited =[0]*len(G)
    dfsindex=[]
    DFSrec(G,v,visited,dfstree,dfsindex)
    return(dfsindex,dfstree)

def ChainDecomp(G):
    index , dfstree = DFS(G,1)

    chains=[]    #List to store chains
    visited={}  #Dictionaries to store the state of vertices
    isinachain={}
    for i in range(1,len(G)+1):
        isinachain[i]=0
        visited[i]=0

    #Traverse vertices in DFS order and looks for backedges
    for i in range(0,len(G)):
        v=index[i]
        visited[v]=1
        for u in G[v]:
            if u==index[0]:
                continue
            if visited[u]==0 and  v!= dfstree[u]:
                visited[u]=1
```

```
                C=[v,u]
                n=dfstree[u]
                while visited[n]==0:
                    C.append(n)
                    visited[n]=1
                    n=dfstree[n]

                C.append(n)

                #This part detects bare paths
                if n!=index[0] and isinachain[n] == 0:
                    n=dfstree[n]
                    while isinachain[n]==0:
                        C.append(n)
                        n=dfstree[n]
                    C.append(n)

                for i in C:
                    isinachain[i]=1
                chains.append(C)
    return(chains)

def intersection(lst1, lst2):
    lst3 = [value for value in lst1 if value in lst2]
    return lst3


def balancedpartition(G):
    V1=[]
    V2=[]
    ch=ChainDecomp(G)

    #Partition C1
    if len(ch[0]) > 4:
        for i in range(0,len(ch[0])-1):
            if i+1 < len(ch[0])/2:
                V1.append(ch[0][i])
            else:
```

```
                V2.append(ch[0][i])
    else:
        for i in range(0,len(ch[0])-1):
            if (ch[0][i]==ch[1][0] or ch[0][i]==ch[1][-1]):
                if len(V1)==0:
                    V1.append(ch[0][i])
                else:
                    V2.append(ch[0][i])


    if len(ch) > 1:
        for i in range(1,len(ch)):
            if ch[i][-1] in V1:
                for j in range(1,len(ch[i])-1):
                    V1.append(ch[i][j])
            else:
                for j in range(1,len(ch[i])-1):
                    V2.append(ch[i][j])


    #Balance
        while abs(len(V1)-len(V2)) > 1:

            if len(V1)<len(V2):
                NV1interV2=[]
                for v in V2:
                    if len(intersection(G[v],V1)) > 0:
                        NV1interV2.append(v)

                for v in NV1interV2:
                    flag=0
                    for w in intersection(G[v],V2):
                        if len(intersection(G[w],V2))==1:
                            flag=1
                    if flag==0:
                        V1.append(v)
                        V2.remove(v)
                        break
```

```

            else:
                NV2interV1=[]
                for v in V1:
                    if len(intersection(G[v],V2)) > 0:
                        NV2interV1.append(v)

                for v in NV2interV1:
                    flag=0
                    for w in intersection(G[v],V1):
                        if len(intersection(G[w],V1))==1:
                            flag=1
                    if flag==0:
                        V2.append(v)
                        V1.remove(v)
                        break

    return(V1,V2)
```