On Threshold Models over Finite Networks

by

Elie M. Adam

B.E., Computer and Communications Engineering
American University of Beirut (2010)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author ....
Department of Electrical Engineering and Computer Science
May 11, 2012

Certified by ....
Munther A. Dahleh
Professor
Thesis Supervisor

Certified by ....
Asuman Ozdaglar
Associate Professor
Thesis Supervisor

Accepted by ....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

On Threshold Models over Finite Networks

by

Elie M. Adam

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

We study a model for cascade effects over finite networks based on a deterministic binary linear threshold model. Our starting point is a networked coordination game where each agent's payoff is the sum of the payoffs coming from pairwise interaction with each of the neighbors. We first establish that the best response dynamics in this networked game is equivalent to the linear threshold dynamics with heterogeneous thresholds over the agents. While the previous literature has studied such linear threshold models under the assumption that each agent may change actions at most once, a study of best response dynamics in such networked games necessitates an analysis that allows for multiple switches in actions. In this thesis, we develop such an analysis and construct a combinatorial framework to understand the behavior of the model. To this end, we establish that the agents behavior cycles among different actions in the limit and provide three sets of results.

We first characterize the limiting behavioral properties of the dynamics. We determine the length of the limit cycles and reveal bounds on the time steps required to reach such cycles for different network structures. We then study the complexity of decision/counting problems that arise within the context. Specifically, we consider the tractability of counting the number of limit cycles and fixed-points, and deciding the reachability of action profiles. We finally propose a measure of network resilience that captures the nature of the involved dynamics. We prove bounds and investigate the resilience of different network structures under this measure.

Thesis Supervisor: Munther A. Dahleh
Title: Professor

Thesis Supervisor: Asuman Ozdaglar
Title: Associate Professor

Acknowledgments

I should start with a note of thanks for my advisors, Prof. Munther Dahleh and Prof. Asu Ozdaglar. Asu, thank you for keeping a keen eye on opportunities, and for your abundant flow of research ideas. You generously offered me your support throughout those two years, and you have tried to help me as much as you can. Munther, thank you for the little tricks I learned from you, on research and quite a few other things as well. You might be surprised and wonder when did I ever get a chance to learn those, but I would say it is the little things that matter most. I still believe that one of the most valuable advice I got from you was "start with the line graph".

I thank all the friends in LIDS, at MIT, outside MIT and far away across the Atlantic. I refrain myself from explicitly mentioning names; forgive me for this. Nevertheless, this is to the friends one sees too often, and for those that time made it as such we separate for a little while. Thank you for the lovely conversations, the comfort and the friendship in dim times.

A special mention goes to Ozan Candogan, Ermin Wei, Annie (I-An) Chen and Kimon Drakopoulos (in desk order) for lavishing our office with the most lovely atmosphere, each adding a spark in his or her unique way. Many thanks for your comfort when some things were just not going the way they should have been. Thank you for your so many advices and for the hearty laughs.

A different thanks goes to (Dr.) Mesrob Ohannessian for his wisest words. I used to climb up the spiral stairs of the 6th floor, and stroll towards your office asking your advice (even after you changed your office). I wish to still do so whenever I can, though you will be setting out soon. I wish you the warmest luck in all.

Whisking back to the Mediterranean coast, I extend my thanks to Prof. Ibrahim Abou-Faycal, Prof. Louay Bazzi and Prof. Fadi Karameh (at the American University of Beirut) for being excellent teachers and friends. Each of you helped me in so many different ways. I strongly believe you could find bits and pieces of what you have taught me scattered along this thesis. So many of those pieces are not just in content.

To my family here, in Massachusetts and outside, thank you for never making me feel so away from Home. Thank you for the wonderful gatherings and the warmest holidays. Uncle George, thank you for helping me settle down in Massachusetts.

To my family not here—specifically Mom and Dad, Karen and Marc—it is always a joy to hear your voices. Thank you for your advice on so many different topics (on which many posed problems had sometimes trivial silly answers). Thank you for enduring with me and supporting me. Thank you for simply being there.

Finally, Romy, thank you for being an angel. You were along my side through it all.

This research was partially supported by an MIT Jacobs Presidential Fellowship, a Siebel Scholarship, AFOSR grant FA9550-09-1-0420, and ARO grant W911NF-09-1-0556.

Contents

List of Figures

Chapter 1

Introduction

1.1 Cascade Effects in Networks

Networks intertwine with every aspect of our modern lives, be it through sharing ideas, communicating information, shaping opinions, performing transactions or delivering utilities. Explicitly, we may cite social networks, financial networks, economic networks, communication networks and power networks. Isolation among entities in many such aggregate systems is being dissolved; more and more links are being established for either mutual or global welfare. However, those interdependencies lay down pathways for various local fluctuations to ripple through. For instance, the repercussion of a firm's collapse echoes among its suppliers, potentially inducing them to also fail if the shocks are to be vigorous enough. If that should happen, it would, in turn, cripple the pool of rivals sharing those same suppliers, and would furthermore 'back-propagate' along the supply chain, eventually crumbling down a whole sector. Similarly, the breakdown of a vital component in a power-grid system (or communication network) might incur enough strain on other components to have them sink, generating an avalanche of successive breakdowns leading to a well-spread outage. We also observe such typical behavior in social settings, such instances hold in adoption of new trends. A new idea sprouts with a small group of early adopters, and is passed on progressively to others by many different means. If conditions happen to be favorable, the idea does not die out but rather builds up enough inertia to flood the whole social network. In the case of infectious diseases, it is quite urgent to contain the propagation, to attempt to limit the contagion process before the disease becomes untameable and takes hold of a favorable portion of the network. Gaining understanding and maneuverability over such processes cannot prove to be but useful, not to mention interesting mathematically in its own right. We need insight and tools to engineer such problems, to construct more resilient networks, mitigate losses and achieve higher efficiency.

1.2 Threshold Models

Interactions over many different types of networks require agents to coordinate with their neighbors. In economic networks, technologies that conform to the standards used by other related firms are more productive; in social networks, conformity to the behavior of friends is valuable for a variety of reasons. The desire for such coordination can lead to cascading behavior: the adoption decision of some agents can spread to their neighbors and from there to the rest of the network. One of the most commonly used models of such cascading behavior is the linear threshold model originally introduced by Granovetter [1]. This model is used to explain a variety of aggregate level behaviors including diffusion of innovation, voting, propagation of rumors and diseases, spread of riots and strikes, and dynamics of opinions.

Most analyses of this model in the literature assume that one of the behaviors adopted by the agents (represented by the nodes of a graph) is irreversible, meaning that agents can only make a single switch into this behavior and can never switch out from it. However, incurring this progressive property in behavior dilutes several perspectives of the dynamics: whereas some situations are best captured by such a variant, many others cannot be captured but by allowing players to revert back to previous actions. A main motivation for example would be opinion dynamics in social network: in most situations a player changes opinions back and forth. This said, the literature lacks a satisfactory characterization of the limiting properties of such a model.

In this thesis, we consider a model of cascade effects based on binary linear threshold dynamics over finite graphs. We start from an explicit coordination game set over a finite undirected network. The payoff of each agent is the sum of the payoff in a *two player and two action coordination game* the agent plays pairwise with each of the neighbors (the action is fixed across all interactions). We then study the behavior induced by best response dynamics, whereby each agent changes the played action to that which yields highest payoff given the actions of the neighbors. We first show that best response dynamics are identical to the dynamics traced by the linear threshold model with heterogeneous thresholds for the agents. However, crucially, actions can change multiple times. Thus, the dynamics of interest for the set of problems posed here cannot be studied using existing results and in fact have a different mathematical structure. The main contribution of this thesis is to fully characterize these dynamics.

Of central importance in the study of cascades over networks is the resilience of networks to invasion by certain types of behavior (e.g., cascades of failures or spread of epidemics). For the new dynamics defined by our problem, we define a measure of resilience of a network to such invasion that captures the heterogeneity in the thresholds of the agents. We prove both upper and lower bounds on the resilience measure, and provide insight on how different network structures affect this measure.

1.3 On Related Literature

The thesis is related to a large literature on network dynamics and linear threshold models (see e.g., [2]-[8]). A number of papers in this literature investigate the question of whether a behavior initially adopted by a subset of agents (i.e., *the seed set*) will spread to a *large* portion of the network, focusing on the dynamics where agents can make a single switch to one of the behaviors. Morris [2], while starting from a multi-switch version of the dynamics, studied without loss of generality the single-switch version to answer whether there exists a finite set of initial adopters (in an infinite network with homogeneous thresholds) such that the behavior diffuses to the entire network. In [6], Watts derives conditions for the behavior to spread to a positive fraction of the network (represented by a random graph with given degree distribution) using a branching process analysis. Similarly, Lelarge [7] provides an explicit characterization of the expected fraction of the agents that adopt the behavior in the limit over such networks.

The work [4] studies the linear threshold model over deterministic graphs. Given an initial seed set of adopter, it characterizes the final set of adopter in terms of *cohesive sets* where cohesion in social groups is measured by comparing the relative frequency of ties among group members to ties with non-members. The work in [5] studies (e.g. in the context of viral marketing) how to target a fixed number of agents (and change their behavior) in order to maximize the spread of the behavior in the network in the (time) limit. Formally, it studies the (optimization) problem of maximizing the final set of adopter, under the constraint of picking K initial adopters. It considers various models of cascade, shows that the optimization problem is NP-hard for the linear threshold model,and then provides an algorithm to find a $(1-1/e)$-approximation for the optimal set that achieves maximum influence.

In the context of network resilience, the recent paper [8] adopts single-switch linear threshold dynamics as a model of failures in a network. This work defines a measure of network resilience that is a function of the graph topology and the distribution over thresholds and studies this measure for different network structures focusing on d-regular graphs (hence ignoring the effect of the degree distribution of a graph on cascaded failures). Here we provide a novel resilience measure that highlights the impact of heterogeneity in thresholds and degrees of different agents.

Finally, noisy versions of best-response dynamics in networked coordination games were studied in [9] and [10] (see also [11] and [12] in the statistical mechanics literature). The random dynamics in these models can be represented in terms of Markov chains with absorbing states, and therefore do not exhibit the cyclic behavior predicted by the multi-switch linear threshold model studied in this thesis.

1.4 Outline and Overview of Contribution

To conclude this chapter, we outline the structure of the thesis and highlight the main results and contribution.

Chapter 2 presents a formal description of the model dynamics as a networked coordination game. We establish that the best-response dynamics along the game are equivalent to the dynamics of a linear threshold model with heterogeneous thresholds distributed over the agents and equal weights on all edges in the graph. We then propose an extension for the model by allowing non-equal weights on the edges, and finally allow self-loops generalizing the graph structure from simple graphs to multi-graphs. Results throughout this thesis will be first established for the primary model, and then generalized to the extension models whenever possible.

The work proceeds in **Chapter 3** to describe the dynamics broadly. We establish that for any network structure, after some finite time step, the agents deterministically cycle among action profiles (we refer to such cycles as convergence cycles). We begin by presenting some global properties of the model, then proceed to characterize limiting properties in the three chapters to follow. We first determine the length of the convergence cycles, we then study the convergence time, i.e., the minimal number of time steps needed to reach a convergence cycle, and finally study the number of cycles and fixed points.

In **Chapter 4**, we characterize the length of the convergence cycles. Ultimately, we show that for any graph structure on the players, any threshold distribution over the players and any initial action configuration played by the players, the limiting behavior of the dynamics get absorbed into action configuration cycles of length at most two. In other words, at the limit every agent either plays one action, never deviating, or keeps on switching actions at every time step. We take care to build up the intuition of the reader by considering specialized instances: we characterize the length of convergence cycles for cycle graphs, complete graphs and trees by exploiting the graph structure. Along the way, we slowly construct the combinatorial framework that is to be used throughout the thesis on general graphs, and emphasize the properties of the constructs to ensure a thorough understanding. We generalize the results to general graphs, extend the results to the case allowing non-equal weights on edges and finally show that the same result holds for multigraphs.

In **Chapter 5**, we characterize the convergence time. Building up on the framework set up in Chapter 4, we show that for some positive integer m, given any graph structure on the players, any threshold distribution over the players and any initial action configuration, the dynamics reach a non-degenerate cycle or a fixed-point in at most mn^2 time steps where n is the number of players. We mention however that similar results on cycle length and quadratic convergence time for linear threshold models (termed differently, e.g. *boolean threshold networks*) have appeared in the Cellular Automata literature in [13]. Nevertheless, we prove our results from a different approach; we put particular emphasis on the combinatorial aspect of the problem. We furthermore improve the convergence time bound from quadratic to be uniformly not more than the size of the network whenever the graph in concern is either a cycle graph, a complete graph or a tree. We finally discuss improving convergence time bounds for the general graph.

Chapter 6 studies the complexity of counting and decision problems that arise in this model. We are interested in characterizing the number of limiting states the system could get absorbed in. We begin by arguing that no 'insightful' uniform upper-bound or lower-bound can be established. Considering only the case of a cycle graph, the number of fixed-points may vary at least from 2 to $2^{n/3}$ depending on the threshold distribution. Instead, we turn to study how tractable it is to count the convergence cycles. We start by an overview of complexity theory, and then proceed to show that:

- Given a graph structure on the player and a threshold distribution over the players as input, the problem of counting the number of limiting configuration classes (i.e. either fixed-points or non-degenerate cycles) is #P-Complete.

- Given a graph structure on the player and a threshold distribution over the players as input, the problem of counting the number of fixed-points is #P-Complete.
- Given a graph structure on the player and a threshold distribution over the players as input, the problem of counting the number of non-degenerate cycles is #P-Complete.

We additionally show that all those counting problems remain #P-Complete even when we restrict the counting to bipartite graphs and homogeneous thresholds. However, restriction to specific graph structures yield counting problems in the complexity class FP: we consider complete graphs as an explicit example. We further consider the problem of deciding whether an action configuration over the network is reachable along the dynamics. To this end, we show that:

- Given a graph structure on the player, a threshold distribution over the players and some action configuration played by the players, the problem of deciding whether that action configuration is reachable is NP-Complete.
- Given a graph structure on the player, a threshold distribution over the players and some *reachable* action configuration played by the players, the problem of counting the number of action configuration preceding that reachable action configuration is #P-Complete.

In **Chapter 7**, we put our model within a context of network resilience. We define a resilience metric as a function of the graph structure that captures the minimal 'cost of recovery' needed when the model is confronted with a perturbation in the agents' action profile. We prove achievable uniform lower-bounds and upper-bounds on the measure. We compute the resilience measure of some network structures, and end with a discussion.

We conclude the thesis with **Chapter 8**.

Chapter 2

Our Model

The model chapter consists of three parts. The first sets up a networked coordination game, it properly defines the interaction among players in the network and characterizes the best-response dynamics in this game. The second proposes a natural extension to the model: it imposes non-equal weights on the pairwise interactions among the players. The third allows players to give weights to their own actions, thus generalizing simple graph structures to multigraph structures.

2.1 The Primary Model

We define a networked coordination game. For a positive integer n, we denote by $\mathcal{I}_n$ the set of n players[1]. For technical convenience, we assume that $\mathcal{I}_n \subset \mathcal{I}_m$ for $n < m$.[2] We define $\mathcal{G}_n$ to be the class of all connected undirected graphs $G(\mathcal{I}_n, E)$ defined over the vertex set $\mathcal{I}_n$, with edge set E.[3] To be proper, E is a relation[4] on $\mathcal{I}_n$, but for convenience we will consider the set E to have cardinality exactly equal

[1]We use the words player, agent, node and vertex interchangeably.

[2]We use the letters i and j to denote agents. We reserve the letter n for the number of players in the game. If it is clear from the context to which set X an element x belongs to, we refrain from mentioning the set X explicitly to simplify notation. Moreover, for any function f with domain $\mathcal{I}_n$, we will denote $f(i)$ by f_i. In particular, for functions q, k and a with domain $\mathcal{I}_n$, $q(i)$, $k(i)$ and $a(i)$ are denoted q_i, k_i and a_i respectively.

[3]For a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and edge set respectively.

[4]A (binary) relation R on a set A is a subset of $A \times A$. We use the notation aRb to denote $(a, b) \in R$.

to the number of undirected edges. We denote an undirected edge in E by $\{i,j\}$, and we abbreviate it to ij when no confusion arises. For $G(\mathcal{I}_n, E)$ in $\mathcal{G}_n$, we use $\mathcal{N}_G(i)$ to denote the neighborhood of player i in G, i.e. $\mathcal{N}_G(i) = \{j \in \mathcal{I}_n : ij \in E\}$. We denote by $d_G(i)$ the degree of player i in G, namely the cardinality of $\mathcal{N}_G(i)$. We refer to $\mathcal{N}_G(i)$ and $d_G(i)$ respectively as $\mathcal{N}_i$ and d_i when the underlying graph is clear from the context. We finally define $\mathcal{Q}_n$ to be the space of type distributions over the agents, namely the set of maps from $\mathcal{I}_n$ into $[0,1]$.

Let $\{\mathbb{B}, \mathbb{W}\}$ be a (binary) set of actions, where the symbols $\mathbb{B}$ and $\mathbb{W}$ may be identified with the colors *black* and *white*, respectively. Given a graph $G(\mathcal{I}_n, E)$ in $\mathcal{G}_n$ and a type distribution q in $\mathcal{Q}_n$, each player i in $\mathcal{I}_n$ plays one action a_i in $\{\mathbb{B}, \mathbb{W}\}$. For $ij \in E$, we define the payoff received by agent i when playing a_i against agent j playing a_j to be

$$g_{i,j}(a_i, a_j) = \begin{cases} q_i & \text{if } a_i = a_j = \mathbb{W} \\ 1 - q_i & \text{if } a_i = a_j = \mathbb{B} \\ 0 & \text{if } a_i \neq a_j \end{cases}. \tag{2.1}$$

The utility player i gets is the sum of the payoffs from the pairwise interactions with the players in $\mathcal{N}_i$. Formally, when player j plays action a_j, we have:

$$u_i(a_i, a_{-i}) = \sum_{j \in \mathcal{N}_i} g_{i,j}(a_i, a_j), \tag{2.2}$$

where a_{-i} denotes the action profile of all players except i.

We define $\mathcal{A}_n$ be the space of action[5] profiles[6] played by the agents, namely the set of maps from $\mathcal{I}_n$ into $\{\mathbb{B}, \mathbb{W}\}$. The players are assigned an initial action profile $\underline{a}$, we refer to $\underline{a}$ as the action profile of the players at time step 0. For T in $\mathbb{N}$,[7] every player best responds to the action profile of the players at time step $T-1$, by choosing the action that maximizes his utility. We suppose that players play action $\mathbb{W}$ as a tie breaking rule. Formally we impose a strict order on $\{\mathbb{W}, \mathbb{B}\}$ such that $\min\{\mathbb{W}, \mathbb{B}\} = \mathbb{W}$. Suppose we denote by $a_{i,T}$ the action played by player i at time T,

[5] We use the words action, assignment and color interchangeably.
[6] We use the words profile and configuration interchangeably.
[7] We denote by $\mathbb{N}$ the set of non-negative integers, and by $\mathbb{Z}^+$ the set of positive integers.

then given an initial action configuration $\underline{a}$ in $\mathcal{A}_n$, for every player i, we recursively define

$$\begin{aligned} a_{i,0} &= \underline{a}_i \\ a_{i,T} &= \min \underset{a_i \in \{\mathbb{W},\mathbb{B}\}}{\operatorname{argmax}} u_i(a_i, a_{-i,T-1}), \quad \text{for } T \in \mathbb{Z}^+. \end{aligned} \tag{2.3}$$

where the min operator breaks ties. The recursive definition in (2.3) is equivalent to the following proposition.

Proposition 2.1.1. *Let $\underline{a}$ be the initial action configuration, namely the action profile of the players at time step 0. For every positive integer T, player i plays action $\mathbb{B}$ at time step T if and only if more than $q_i d_i$ neighbors of player i played action $\mathbb{B}$ at time step $T-1$.*

Proof. We substitute u_i in (2.3) with the expressions in (2.1) and (2.2), and get that player i plays action $\mathbb{B}$ at time T if and only if

$$\sum_{j \in \mathcal{N}_i} (1 - q_i) \mathbf{1}_{\{\mathbb{B}\}}(a_{j,T-1}) > \sum_{j \in \mathcal{N}_i} q_i \mathbf{1}_{\{\mathbb{W}\}}(a_{j,T-1}),$$

where $\mathbf{1}_\Gamma(x) = 1$ if and only if $x \in \Gamma$. Equivalently, player i plays action $\mathbb{B}$ at time T if and only if

$$\sum_{j \in \mathcal{N}_i} \mathbf{1}_{\{\mathbb{B}\}}(a_{j,T-1}) > q_i d_i.$$

The left-side term is essentially summing the number of neighbors of player i playing action $\mathbb{B}$. $\square$

As a technical clarification, we highlight the fact that every player is capable of switching actions both from $\mathbb{W}$ to $\mathbb{B}$ and $\mathbb{B}$ to $\mathbb{W}$.

2.2 An Extension: Dynamics with Weighted Edges

Our primary model is such that every player treats the payoffs from the pairwise interactions with equal weights. This corresponds on the part of player i to an unweighted counting of the number of neighbors playing $\mathbb{B}$ at time $T-1$ to decide whether to play $\mathbb{B}$ at T or not. Our model can take a more general form by allowing symmetric positive weights on the payoffs. Let $G(\mathcal{I}_n, E)$ be given, suppose we assign for every ij in E, a positive weight $w_{ij} = w_{ji}$. Given a q in $\mathcal{Q}_n$, we extend the utility player i gets to be the weighted sum of the payoffs from the pairwise interactions with the players in $\mathcal{N}_i$, namely when player j plays action a_j,

$$u_i(a_i, a_{-i}) = \sum_{j \in \mathcal{N}_i} w_{ij} g_{i,j}(a_i, a_j), \tag{2.4}$$

Again, we denote by $a_{i,T}$ the action played by player i at time T. If we let $\underline{a}$ be the initial action configuration, namely $a_{i,0} = \underline{a}_i$, then for every positive integer T, player i plays action $a_{i,T} = \mathbb{B}$ at time step T if and only if

$$\sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{1}_{\{\mathbb{B}\}}(a_{j,T-1}) > \theta_i,$$

where we define $\theta_i = q_i \sum_{j \in \mathcal{N}_i} w_{ij}$. The primary model described in the previous section is then an instance of this model where $w_{ij} = w_{ji} = 1$ for all edges ij in E. This model translates to allowing multiple edges among nodes in the primary model.

2.3 An Extension: Dynamics over Multigraphs

In both the primary model and its weighted-edge extension, a node does not take into consideration the action it played at time step $T-1$ when playing an action at time step T. We may go around this issue by allowing the node to play a coordination game with itself. This translates to allowing weighted self loops in the network structure. Combining both weights on the self loops and on the edge leads to a graph structure that is not necessarily a simple graph but rather a multigraph.

Definition 2.3.1. *A multigraph is a pair of a set V and a multiset*[8] *E having a subset*

[8] The notion of a multiset generalizes the notion of a set by allowing elements to appear multiple times. Formally, a multiset is a set S along with a map m from S into $\mathbb{Z}^+$, where for each s in S, the positive integer $m(s)$ denotes the multiplicity of s. We shall refer to S as the base set.

of $V \times V$ as a base set. The set V is called the vertex set; the set E is called the edge set.

The decision rule extends naturally from the previous models by letting each node suppose itself as a neighbor with a symmetrically weighted edge. We refrain from explicitly writing the update rule.

The Course of the Thesis will proceed as follows. Much emphasis will be given to extracting dynamical properties of the primary model described in Section 2.1. Those properties will be furthermore generalized to the model in Section 2.2 and Section 2.3. A general description of the dynamics and a general overview of the results is provided in the next chapter. The resilience context will be considered in the Chapter 8. We will prove bounds on the measure, and investigate different network structures with respect to that measure.

Chapter 3

Description of the Dynamics

We begin with a coarse description of the involved dynamics. However, we only focus on the primary model. The dynamics in the extension models may be trivially generalized from the propositions in this chapter. To sum up the model, we consider a *finite* set of players $\mathcal{I}_n$ along with three mathematical objects $\mathcal{G}_n$, $\mathcal{Q}_n$ and $\mathcal{A}_n$. An element $G(\mathcal{I}_n, E)$ of $\mathcal{G}_n$ corresponds to the network structure imposed on the players, an element q of $\mathcal{Q}_n$ refers to the type distribution over the players, and an element a of $\mathcal{A}_n$ consists an action profile played by the players. The triplet G, q and a interact as dictated by Proposition 2.1.1.

3.1 From Types to Thresholds

Proposition 2.1.1 infers that playing $\mathbb{B}$ is never a best response for player i if no player in $\mathcal{N}_i$ is playing $\mathbb{B}$. We will generalize our model to provide symmetry between both actions $\mathbb{B}$ and $\mathbb{W}$. We do this for two reasons. The first is to consider the linear threshold model as considered in the literature. The second is a technical reason, mainly to ensure closure of the set $\mathcal{G}_n \times \mathcal{Q}_n$ under certain operations. Nevertheless, any result for the generalized version of the model is inherited by the initial version trivially by inclusion.

We substitute the set $\mathcal{Q}_n$ by a set $\mathcal{K}_n$ and then modify the statement of Proposition 2.1.1. We define $\mathcal{K}_n$ to be the space of threshold distributions over the agents, namely the set of maps from $\mathcal{I}_n$ into $\mathbb{N}$. We make a particular distinction between the word *type* attributed to $\mathcal{Q}_n$ and the word *threshold* attributed to $\mathcal{K}_n$. Given a pair (G, k) with $k \in \mathcal{K}_n$, we generalize Proposition 2.1.1 as follows:

Proposition 3.1.1. *Let $\underline{a}$ be the initial action configuration, namely the action profile of the players at time step 0. For every positive integer T, player i plays action $\mathbb{B}$ at time step T if and only if at least k_i neighbors of player i played action $\mathbb{B}$ at time step $T-1$.*

The rule in Proposition 3.1.1 supersets the rule in Proposition 2.1.1. Indeed, for every q in $\mathcal{Q}_n$ there exists a k in $\mathcal{K}_n$ such that $q_i d_i$ may be substituted with the integer k_i for all i without changing the behavior of the players. It is also crucial to note that *at most* is replaced by *at least.*

Having made the transition from *types* to *thresholds*, we distinguish the nodes having thresholds at the *boundaries* as follows:

Definition 3.1.2. *Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$, player i in $\mathcal{I}_n$ is called non-valid with respect to (G, k) (or simply non-valid) if k_i is either equal to 0 or greater than d_i. A player is called valid if it is not non-valid.*

A non-valid node is then allowed to play only one of the actions in $\{\mathbb{W}, \mathbb{B}\}$ (depending on its threshold) whenever it is allowed to decide on the action to play.

Finally, for $G(I_n, E)$ in $\mathcal{G}_n$ and k in $\mathcal{K}_n$, we denote by G_k the map from $\mathcal{A}_n$ into $\mathcal{A}_n$ such that for player i, $(G_k a)_i = \mathbb{B}$ if and only if at least k_i players are in $a^{-1}(\mathbb{B}) \cap \mathcal{N}_i$.[1] From this perspective, given an initial configuration a in $\mathcal{A}_n$, the sequence $a, G_k a, G_k^2 a, \cdots$ corresponds to the sequence of action profiles $a, a_1, a_2, \cdots$ where $a_T = G_k^T a$ is the action profile played by the players at time T if they act in accordance with the rule in Proposition 3.1.1.

[1]Let $f : A \to B$ and $g : B \to C$ be functions, we denote by gf the function $g \circ f : A \to C$. In particular, if a function f maps a set A to itself, for a non-negative integer m, we denote by f^m the function $f \circ f^{m-1}$ where f^0 is the identity map on A.

3.2 The Limiting Behavior

To understand the limiting behavior, we note two fundamental properties: the space $\mathcal{A}_n$ has finite cardinality, and Proposition 3.1.1 is deterministic. Since $\mathcal{A}_n$ is finite, if we let $a_0, a_1, a_2, \cdots$ be any infinite sequence of action profiles played by the agents according to Proposition 3.1.1, then there exists at least one action profile $\hat{a}$ that will appear infinitely many times along this sequence. This follows from the pigeon-hole principle. Since the dynamics are deterministic, the same sequence of action profiles appears between any two consecutive occurrences of $\hat{a}$. This means that after a finite time step, the sequence of action profiles will cycle among action profiles.

Let us consider a different representation of the dynamics. Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$, we define a (binary) relation $\rightarrow$ on $\mathcal{A}_n$ such that for a and b in $\mathcal{A}_n$, $a \rightarrow b$ if and only if $b = G_k a$. Consider the graph $H(\mathcal{A}_n, \rightarrow)$, it forms a directed graph (possibly with self loops) on the vertex set taken to be the space of action profiles $\mathcal{A}_n$, and an action profile a is connected to an action profile b by a directed edge (a, b) going from a to b if and only if $b = G_k a$. Suppose we pick a vertex a, namely an action configuration, and perform a walk on vertices along the edges in H starting from a. The walk eventually cycles vertices in the same order. Every initial action profile leads to one cycle, and two action profiles need not lead to the same cycle. We formalize the idea in the following definitions.

Definition 3.2.1. *Given (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$, for two action profiles a and b in $\mathcal{A}_n$, it is said that a can be reached from b with respect to G_k if there exists a non-negative integer T such that $a = G_k^T b$. Formally, we define the relation $\mathcal{R}_{G_k}$ on $\mathcal{A}_n$ such that for a and b in $\mathcal{A}_n$, $a \mathcal{R}_{G_k} b$ if and only if there exists a non-negative integer T such that $a = G_k^T b$.*

For a and b in $\mathcal{A}_n$, we have $a \mathcal{R}_{G_k} b$ if and only if there exists a directed path in $H(\mathcal{A}_n, \rightarrow)$ from vertex b to vertex a. The idea to emphasize is that H is not necessarily weakly-connected.[2] If we construct a relation $\mathcal{C}_{G_k}$ on $\mathcal{A}_n$ such that for a and b in $\mathcal{A}_n$,

[2] A directed graph G is said to be weakly-connected if for any vertices u and v in the vertex set of G, there exists an *undirected* path connecting u to v. A weakly-connected component of G is a maximal subgraph of G that is weakly-connected.

$a\mathcal{C}_{G_k}b$ if and only if $a\mathcal{R}_{G_k}b$ or $b\mathcal{R}_{G_k}a$, then $\mathcal{C}_{G_k}$ is an equivalence relation on $\mathcal{A}_n$. In this setting, two configurations in $\mathcal{A}_n$ are in the same equivalence class with respect to the relation $\mathcal{C}_{G_k}$ if and only if they are in the same weakly-connected component in H. In this case, every weakly-connected component of H contains exactly one directed cycle. We characterize the set of cycles as follows:

Definition 3.2.2. *Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$, we define $CYCLE_n(G, k)$ to be the collection of subsets of $\mathcal{A}_n$, such that for every C in $CYCLE_n(G, k)$, if a and b are in C then we have both $a\mathcal{R}_{G_k}b$ and $b\mathcal{R}_{G_k}a$, and for every c in $\mathcal{A}_n \backslash C$, there does not exist an action configuration a in C such that $a\mathcal{R}_{G_k}c$. We refer to the elements of $CYCLE_n(G, k)$ as convergence cycles.*

The condition "$a\mathcal{R}_{G_k}b$ and $b\mathcal{R}_{G_k}a$" can be concisely replaced by "$a\mathcal{R}_{G_k}b$", however we keep it as such to stress on the fact that both a can be reached from b and b can be reached from a. The second condition ensures that C is in $CYCLE_n(G, k)$ only if there exists no larger cycle C' containing C.

Cycles in $CYCLE_n(G, k)$ consisting of only one action configuration are fixed-points of G_k and so will be referred to as *fixed-points*. Cycles in $CYCLE_n(G, k)$ consisting of more than one action configuration will be referred to as *non-degenerate cycles* (as opposed to *fixed-points* which are *degenerate cycles*).

We begin by stating some basic properties of the dynamics. First, we define two partial-order relations namely $\mathbb{B}$-inclusion ($\subset_{\mathbb{B}}$) and $\mathbb{W}$-inclusion ($\subset_{\mathbb{W}}$) on $\mathcal{A}_n$.

Definition 3.2.3. *For a and b in $\mathcal{A}_n$, we have $a \subset_{\mathbb{B}} b$ if and only if $a^{-1}(\mathbb{B}) \subset b^{-1}(\mathbb{B})$. Similarly, we have $a \subset_{\mathbb{W}} b$ if and only if $a^{-1}(\mathbb{W}) \subset b^{-1}(\mathbb{W})$.*

From the definition, we have that $a \subset_{\mathbb{B}} b$ if and only if $b \subset_{\mathbb{W}} a$. Building on the definition, the dynamics involved are monotonic in the following sense:

Property 3.2.4 (Monotonicity). *For every pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$, and every a and b in $\mathcal{A}_n$, if $a \subset_{\mathbb{B}} b$ then $G_k a \subset_{\mathbb{B}} G_k b$.*

Proof. For a and b in $\mathcal{A}_n$, if $a \subset_{\mathbb{B}} b$ then for each i in $\mathcal{I}_n$, we have $a^{-1}(\mathbb{B}) \cap \mathcal{N}_i \subset b^{-1}(\mathbb{B}) \cap \mathcal{N}_i$. To finish off the proof, we invoke the decision rule under G_k. Specifically, $(G_k a)_i = \mathbb{B}$ if and only if $|a^{-1}(\mathbb{B}) \cap \mathcal{N}_i| \geq k_i$, and therefore $(G_k a)_i = \mathbb{B}$ only if $|b^{-1}(\mathbb{B}) \cap \mathcal{N}_i| \geq k_i$, which in turn is equivalent to $(G_k b)_i = \mathbb{B}$. □

As a follow-up, we can derive sufficient conditions for the sequence $a, G_k a, G_k^2 a, \cdots$ to be eventually constant i.e. sufficient conditions to attain a fixed-point of G_k when applying G_k iteratively on a finitely many times.

Proposition 3.2.5. *For every pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ and every a in $\mathcal{A}_n$, if either $a \subset_{\mathbb{B}} G_k a$ or $a \subset_{\mathbb{W}} G_k a$ then there exists an action configuration c in $\mathcal{A}_n$ such that $G_k^m a = c$ for all m greater or equal to some non-negative integer M.*

Proof. We consider the case where $a \subset_{\mathbb{B}} G_k a$, the other case may be derived by symmetry. If $a \subset_{\mathbb{B}} G_k a$ then $a \subset_{\mathbb{B}} G_k a \subset_{\mathbb{B}} G_k^2 a \subset_{\mathbb{B}} G_k^3 a \subset_{\mathbb{B}} \cdots$ by monotonicity. Clearly, if $G_k^p a = G_k^{p+1} a$ for some non-negative integer M, then $G_k^p a = G_k^{p+m} a$ for all non-negative integers m, since $G_k^{p+m+1} = G_k G_k^{p+m}$. Finally, by the pigeonhole principle and monotonicity, there exists a non-negative integer M such that $G_k^M a = G_k^{M+1} a$. □

Nevertheless, this condition is not a necessary condition. We provide a simple example to illustrate that fact. We consider the 2-regular connected graph R in $\mathcal{G}_3$ and define k to be equal to 1 over all players in $\mathcal{I}_3$. We pick a player i in $\mathcal{I}_n$ and consider an action configuration a in $\mathcal{A}_3$ to be equal to $\mathbb{B}$ on i and equal to $\mathbb{W}$ everywhere else. It then follows that $G_k a$ and a are not comparable with respect to $\mathbb{B}$-inclusion (and hence also $\mathbb{W}$-inclusion). Indeed, $(G_k a)_i = \mathbb{W}$ and $(G_k a)_j = \mathbb{B}$ for j in $\mathcal{N}_i$. However, $G_k^2 a$ is equal to $\mathbb{B}$ over $\mathcal{I}_n$, and $G_k^2 a = G_k^{2+m} a$ for every non-negative integer m.

Finally, we present a rather natural statement that will be extensively used in inductive arguments. Although the statement is simple, we take effort to state it carefully to invoke it, whenever needed, without having to take care of any minor technicalities involved. In addition, the insight behind the statement is crucial enough to be stated outside a proof. We first give a simplistic non-formal version of it: given a triplet (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$, let us consider the sequence $a, G_k a, G_k a^2, \cdots$. If some player *never* changes action along this sequence, then we may delete the player from the graph and modify the thresholds of the neighboring players in such a way that the effect of the action played by the player is 'seen' by the neighbors. We do so by keeping the thresholds of the neighbors unchanged if that action is $\mathbb{W}$, and by decreasing the thresholds by 1 if that action is $\mathbb{B}$. In this case, we would obtain a different triplet (G', k', a') where all the players in $V(G')$ play the exact actions as in the sequence $a, G_k a, G_k a^2, \cdots$.

We initially present the statement assuming the graph is 2-connected[3] (for simplicity) while omitting the proof. We then follow it with a generalized statement relaxing the 2-connectedness assumption, and provide a proof there instead.

Proposition 3.2.6. *Let (G, k) be a pair in $\mathcal{G}_n \times \mathcal{K}_n$ where G is 2-connected, and let a be an action configuration in $\mathcal{A}_n$. If there exists a player i and an action c, such that $(G_k^m a)_i = c$ for all non-negative integers m, we let G' be the induced subgraph[4] of G over $\mathcal{I}_n \backslash \{i\}$,[5] define a' to be the action configuration a restricted to the players in $\mathcal{I}_n \backslash \{i\}$, and define k' to be the map from $\mathcal{I}_n \backslash \{i\}$ into $\mathbb{N}$ such that $k' = k$ on $\mathcal{I}_n \backslash (\mathcal{N}_i \cup \{i\})$, $k' = k$ on $\mathcal{N}_i$ if $c = \mathbb{W}$ and $k' = (k-1) \vee 0$ on $\mathcal{N}_i$ if $c = \mathbb{B}$.[6] Then,*

$$(G_k^m a)_j = (G_{k'}'^m a')_j,$$

for all non-negative integers m and all players j in $\mathcal{I}_n \backslash \{i\}$.

We have only defined our dynamics over connected connected graphs, and so the 2-connectedness assumption is only needed to ensure that the graph is connected when deleting a vertex. The assumption may be relaxed by restricting the analysis

[3]A connected graph is said to be 2-connected if the graph remains connected when one vertex of the graph is removed.

[4]Let G be a graph, an induced subgraph over $U \subset V(G)$ is the graph defined over the vertex set U with edge set E consisting of the edges in $E(G)$ having both endpoints in U.

[5]Let X be a set. For A and B subsets of X, we denote by $A \backslash B$ the subset of X containing elements in A that are not in B.

[6] For x_1 and x_2 in a strictly ordered set, we denote $\max\{x_1, x_2\}$ and $\min\{x_1, x_2\}$ by $x_1 \vee x_2$ and $x_1 \wedge x_2$ respectively.

in the statement to only a connected component of the induced subgraph.

Proposition 3.2.7. *Let (G, k) be a pair in $\mathcal{G}_n \times \mathcal{K}_n$, and let a be an action configuration in $\mathcal{A}_n$. If there exists a player i and an action c, such that $(G_k^m a)_i = c$ for all non-negative integers m, we let H be the induced subgraph of G over $\mathcal{I}_n \backslash \{i\}$. Suppose G' is a connected component of H with vertex set $\mathcal{J}$, define a' to be the action configuration a restricted to the players in $\mathcal{J}$, and define k' to be the map from $\mathcal{J}$ into $\mathbb{N}$ such that $k' = k$ on $\mathcal{J} \backslash \mathcal{N}_i$, $k' = k$ on $\mathcal{N}_i \cap \mathcal{J}$ if $c = \mathbb{W}$ and $k' = (k-1) \vee 0$ on $\mathcal{N}_i \cap \mathcal{J}$ if $c = \mathbb{B}$. Then,*

$$(G_k^m a)_j = (G'^m_{k'} a')_j,$$

for all non-negative integers m and all players j in $\mathcal{J}$.

Proof. It would be enough to show that the local decision rules of the players in $\mathcal{J} \cap \mathcal{N}_i$ does not change, i.e. for j in $\mathcal{J} \cap \mathcal{N}_i$, we have:

$$(G_k a)_j = (G'_{k'} a')_j.$$

Let j be a player in $\mathcal{J} \cap \mathcal{N}_i$, then $(G_k a)_j = \mathbb{B}$ if and only if at least k_j nodes in $\mathcal{N}_j$ play $\mathbb{B}$ in a, or equivalently at least k'_j nodes in $\mathcal{N}_j \backslash \{i\}$ play $\mathbb{B}$ in a', since k' takes into account the action of player i. But the last statement is equivalent to $(G'_{k'} a')_j = \mathbb{B}$. $\square$

Note: Non-valid nodes may be created during this deletion process, even when starting with only valid nodes. From this perspective, the class of elements (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ such that all nodes are valid with respect to (G, k) is not closed under this deletion operation. Care should be taken when performing proofs by induction based on node deletion.

We proceed to provide a broad overview of all the main general results, while omitting the proofs. We elaborate on each result along with refinements in subsequent chapter. We restrict the result to those regarding the behavior of the dynamics. We do not mention any resilience measure or bounds thereof, we do so in Chapter 8.

3.3 On Convergence: an Overview

Given the limiting cyclic behavior, the most natural starting point should characterize the length of the cycles in the equivalence classes as a function of the imposed graph structure and the threshold distribution.

Theorem 3.3.1. *For every positive integer n, every (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ and every cycle C in $CYCLE_n(G, k)$, the cardinality of C is less than or equal to 2.*

Put differently, given a network structure G, a threshold distribution k and an initial action profile a, if we iteratively apply G_k on a ad infinitum to get a sequence of best response action profiles, along the sequence of actions considered by player i, player i will eventually either settle on playing one action, or switch action on every new application of G_k. We further show that this theorem also holds for multigraphs as network structures (see Sections 2.2 and 2.3).

Definition 3.3.2. *For every positive integer n, and every (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$, we define $\delta_n(G, k, a)$ to be equal to the smallest non-negative integer T such that there exists a cycle C in $CYCLE_n(G, k)$ and b in C with $G_k^T a = b$.*

The quantity $\delta_n(G, k, a)$ denotes to the minimal number of iterations needed until a given action configuration a reaches a cycle, when iteratively applying G_k. We refer to $\delta_n(G, k, a)$ as the *convergence time* from a under G_k.

Theorem 3.3.3. *For some positive integer m, every positive integer n, and every (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$, the convergence time $\delta_n(G, k, a)$ is less than or equal to mn^2.*

We mention that similar results on convergence cycles and quadratic convergence time have appeared in the literature on cellular automata [13]. Nevertheless, the proof approach is different: we focus throughout the thesis on building a combinatorial framework for the analysis. Moreover, we improve the results on convergence time and get a bound that is linear in the size of the network when the graphs are restricted to being cycle graphs, complete graphs or trees.

Theorem 3.3.4. *For all positive integers n, and every (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ where G is a cycle graph, a complete graph or a tree, the convergence time $\delta_n(G, k, a)$ is less than or equal to n.*

We further discuss improving convergence time bounds on general graphs.

We proceed to present an overview of results on decision and counting problem that arise within this framework.

3.4 On Complexity: an Overview

A next natural step is to quantify the number of limiting configurations. We characterize the number of equivalence classes, fixed points and cycles of length two (referred to as non-degenerate cycles). We refrain from defining complexity classes in this section, instead we refer the reader to Chapter 6 where we provide a short overview of Complexity Theory. We consider the counting problem #CYCLE that takes $< n, G, k >$ as input, where n is a positive integer and (G, k) belongs to $\mathcal{G}_n \times \mathcal{K}_n$, and outputs the cardinality of $CYCLE_n(G, k)$.

Theorem 3.4.1. *#CYCLE is #P-Complete.*

One has to be subtle towards what such result entails. This result does not imply that no characterization of the number of cycles is possible whatsoever, but rather that we would be unable to get an arbitrarily refined characterization of that number.

We consider the counting problem #FIX that takes $< n, G, k >$ as input, where n is a positive integer and (G, k) belongs to $\mathcal{G}_n \times \mathcal{K}_n$, and outputs the cardinality of $\{C \in CYCLE_n(G, k) : |C| = 1\}$.

Theorem 3.4.2. *#FIX is #P-Complete.*

We consider the counting problem #2CYCLE that takes $< n, G, k >$ as input, where n is a positive integer and (G, k) belongs to $\mathcal{G}_n \times \mathcal{K}_n$, and outputs the cardinality of $\{C \in CYCLE_n(G, k) : |C| = 2\}$.

Theorem 3.4.3. *#2CYCLE is #P-Complete.*

We further show that those counting problems remain hard even if we restrict the graphs to be bipartite and impose homogeneous thresholds on the players.

A question of interest is to decide whether given a graph structure G, a type distribution k and some action configuration a, the action configuration a is reachable from some configuration b. We define the language PRED to consist of all 4-tuples $< n, G, k, a >$, where n is a positive integer, (G, k, a) belongs to $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ with $G_k(a)^{-1} \neq \emptyset$.

Theorem 3.4.4. *PRED is NP-Complete.*

Given a graph structure G, a type distribution k and a configuration a, suppose we want to compute the number of configurations b from which a can be reached by applying G_k only once on b. We define the counting problem $\#PRED$ takes $< n, G, k, a >$ as input, where n is a positive integer and (G, k, a) and outputs the cardinality of $G_k^{-1}(a)$. As a corollary from the hardness of PRED, we get:

Corollary 3.4.5. *$\#PRED$ is #P-Complete.*

However, suppose that we restrict the counting to only the action configurations that are reachable from some action configuration. Specifically, we restrict the counting to only the elements in $PRED$. From this perspective, we are computing the 'fan-in' of a given action configuration.

If we define the counting problem #reachable-PRED to take $< n, G, k, a >$ as input, where n is a positive integer, (G, k, a) and $G_k(a)^{-1} \neq \emptyset$ and output the cardinality of $G_k^{-1}(a)$. We get the following result:

Theorem 3.4.6. *#reachable-PRED is #P-Complete.*

The results are derived from thresholds in $\mathcal{K}_n$ instead of types of $\mathcal{Q}_n$. However, the results trivially extend to types as follows: Convergence results hold by inclusion;

complexity results hold since they still hold if we restrict (G, k) to contain no non-valid node. We devote the Chapters 4 and 5 to follow to convergence results, and Chapter 6 to complexity results.

Chapter 4

On Convergence Cycles

In this chapter, we study the following problem: given a graph G in $\mathcal{G}_n$ and a threshold distribution k in $\mathcal{K}_n$, how many action configurations could a cycle in $CYCLE_n(G, k)$ consist of? Ultimately, we show that for any graph and any threshold distribution, the cycles in $CYCLE_n(G, k)$ consist of at most two action configurations. We begin the analysis by considering cycle graphs, we proceed to complete graphs and then move on to trees. Each of those special cases is treated by exploiting its graphic properties. Obviously, most of those properties are not shared among all graphs, and some cannot even be generalized to general graphs. Nevertheless, we explicitly provide results over those toy examples first to build up the intuition of the reader and second to construct the combinatorial framework slowly as we go along. After trees, we consider general graphs. We then generalize the results to the extension model allowing non-equal weights on edges and finally extend the results over multigraphs as network structure.

4.1 On Convergence Cycles for Cycle Graphs

[1]

Let us consider a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is a path, i.e. every agent is connected to at most two other agents and no cycles in the graphs are allowed. Our

[1] A cycle graph is a 2-regular connected graph, we shall use both terms interchangeably.

first intent is to characterize the length of the limiting cycles in that case. Suppose k is picked in such a way that some players are non-valid, then we know that those players can only play one action after some finite time step. With respect to the analysis concerned, we may remove those players as in Proposition 3.2.7, update the thresholds of the neighboring players accordingly and end up with a collection of disconnected paths. Restricting the analysis to one of the paths leads us back to the initial case. Therefore, we will assume that every node in the graph is valid, this implies that k is equal to 1 for the nodes having degree 1 and k takes values in $\{1, 2\}$ for the nodes having degree equal to 2. Moreover, to take care of the boundary case, we will connect the 1-degree nodes together, and so forming a ring of agents. The graph in consideration is then the 2-regular connected graph. We then relax k to take values in $\{1, 2\}$ over $\mathcal{I}_n$.

Given that the thresholds of the nodes are either 1 or 2, it is interesting to state the decision rules as follows. Let a be some action configuration in $\mathcal{A}_n$, if node i has a threshold k_i equal to 1, then node i is $\mathbb{B}$ in $G_k a$ if and only if either one of its neighbors is $\mathbb{B}$ in a. Similarly, if node i has a threshold k_i equal to 2, then node i is $\mathbb{B}$ in $G_k a$ if and only if both of its neighbors are $\mathbb{B}$ in a.

Let us impose a strict ordering on $\{\mathbb{W}, \mathbb{B}\}$ such that $\min\{\mathbb{W}, \mathbb{B}\} = \mathbb{W}$. This translates to $\mathbb{W} \wedge \mathbb{B} = \mathbb{W}$ for notational convenience.[2] For maps a, b and c taking values in $\{\mathbb{W}, \mathbb{B}\}$, the following identities can be checked that:

$$\begin{aligned} a \wedge a &= a & a \vee a &= a \\ a \wedge \mathbb{B} &= a & a \vee \mathbb{W} &= a \\ a \wedge b &= b \wedge a & a \vee b &= b \vee a \\ a \wedge (b \vee c) &= (a \wedge b) \vee (a \wedge c) & a \vee (b \wedge c) &= (a \vee b) \wedge (a \vee c) \\ a \wedge (b \wedge c) &= (a \wedge b) \wedge c & a \vee (b \vee c) &= (a \vee b) \vee c. \end{aligned}$$

Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is 2-regular, all nodes are valid in (G, k)

[2] See footnote 6. Mainly, $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$.

and k takes values in $\{1,2\}$, we define a map τ from $\mathcal{I}_n$ into $\{\vee, \wedge\}$ such that $\tau_i = \vee$ if and only if $k_i = 1$. Similarly, let us define two maps s and p from $\mathcal{I}_n$ into $\mathcal{I}_n$ (we refer to them successor and predecessor) such that i and s_i are neighbors, i and p_i are neighbors and $(sp)_i = (ps)_i = i$. The dynamics are then represented as follows:

$$(G_k a)_i = a_{p_i} \tau_i a_{s_i}.$$

We give a quick example to illustrate. Let us consider a 2-regular connected graph G over the set $\mathcal{I}_n$ for $n > 5$ and suppose we are given a threshold distribution k in $\mathcal{K}_n$ taking values in $\{1,2\}$. Let us choose a node i from $\mathcal{I}_n$. The nodes s_i and p_i are then (distinct) neighbors of node i. Let a be an action configuration in $\mathcal{A}_n$ and suppose $k_i = 1$, then $(G_k a)_i = \mathbb{B}$ if and only if either $a_{s_i} = \mathbb{B}$ or $a_{p_i} = \mathbb{B}$ i.e. at least one neighbor is $\mathbb{B}$. We can rewrite the previous statement as:

$$(G_k a)_i = \max\{a_{s_i}, a_{p_i}\} = a_{s_i} \vee a_{p_i} = a_{s_i} \tau_i a_{p_i},$$

that is because we imposed a strict ordering on $\{\mathbb{W}, \mathbb{B}\}$ such that $\min\{\mathbb{W}, \mathbb{B}\} = \mathbb{W}$, and because $\tau_i = \vee$ if and only if $k_i = 1$ by definition. Furthermore, s_i has both i and $(ss)_i$ as neighbors. Suppose $k_{s_i} = 2$, then in this case $(G_k a)_{s_i} = \mathbb{B}$ if and only if both $a_i = \mathbb{B}$ and $a_{(ss)_i} = \mathbb{B}$ i.e. at least two neighbor are $\mathbb{B}$. Similarly, we can rewrite the previous statement as:

$$(G_k a)_i = \min\{a_i, a_{(ss)_i}\} = a_i \wedge a_{(ss)_i} = a_i \tau_{s_i} a_{(ss)_i}.$$

Finally to conclude the example, we further suppose that $k_{p_i} = 2$. The node p_i has both i and $(pp)_i$ as neighbors, and similarly to the rule of node s_i, we have $(G_k a)_i = a_i \wedge a_{(pp)_i}$. We may now express $(G_k^2 a)_i$ in terms of actions in a as follows:

$$(G_k^2 a)_i = (G_k(G_k a))_i = (G_k a)_{s_i} \vee (G_k a)_{p_i} = (a_i \wedge a_{(ss)_i}) \vee (a_i \wedge a_{(pp)_i}).$$

Finally, using distributivity (as defined in the identities above) we get:

$$(a_i \wedge a_{(ss)_i}) \vee (a_i \wedge a_{(pp)_i}) = a_i \wedge (a_{(ss)_i} \vee a_{(pp)_i}).$$

We now generalize the last part of the example. We consider a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is 2-regular and k takes values in $\{1, 2\}$. In this setting, $(G_k^2 a)_i$ would only depend on the actions of nodes $(ss)_i$, $(pp)_i$ and i itself in a. In particular, there are a total of eight possible decision rules, we summarize them in the following table:

	τ_{p_i}	τ_i	τ_{s_i}	$(G_k a)_i$	$(G_k^2 a)_i$
c.1	$\vee$	$\vee$	$\vee$	$a_{p_i} \vee a_{s_i}$	$a_i \vee (a_{(ss)_i} \vee a_{(pp)_i})$
c.2	$\vee$	$\vee$	$\wedge$	$a_{p_i} \vee a_{s_i}$	$a_i \vee a_{(pp)_i}$
c.3	$\vee$	$\wedge$	$\vee$	$a_{p_i} \wedge a_{s_i}$	$a_i \vee (a_{(ss)_i} \wedge a_{(pp)_i})$
c.4	$\vee$	$\wedge$	$\wedge$	$a_{p_i} \wedge a_{s_i}$	$a_i \wedge a_{(ss)_i}$
c.5	$\wedge$	$\vee$	$\vee$	$a_{p_i} \vee a_{s_i}$	$a_i \vee a_{(ss)_i}$
c.6	$\wedge$	$\vee$	$\wedge$	$a_{p_i} \vee a_{s_i}$	$a_i \wedge (a_{(ss)_i} \vee a_{(pp)_i})$
c.7	$\wedge$	$\wedge$	$\vee$	$a_{p_i} \wedge a_{s_i}$	$a_i \wedge a_{(pp)_i}$
c.8	$\wedge$	$\wedge$	$\wedge$	$a_{p_i} \wedge a_{s_i}$	$a_i \wedge (a_{(ss)_i} \wedge a_{(pp)_i})$

We proceed by defining strong assignments, then state a first proposition on the dynamics induced by the above table.

Definition 4.1.1. *For any positive even integer n greater than 2, any graph G in $\mathcal{G}_n$, every threshold distribution k in $\mathcal{K}_n$, an action c is called a strong action (or strong assignment) for player i in $\mathcal{I}_n$ if once played by player i at time step T, it is played by player i at time step $T + 2m$ for all positive integers m, regardless of what is initially played by the neighbors of node i. We refer to any action that is not a strong assignment as a weak assignment.*

Given the definition, every node in the setting concerned in this section has a strong assignment.

Proposition 4.1.2. *For any positive even integer n greater than 2, any 2-regular graph G in $\mathcal{G}_n$, any threshold distribution k in $\mathcal{K}_n$ taking values in $\{1,2\}$, every player i in $\mathcal{I}_n$ has a strong assignment.*

Proof. Let i be a player in $\mathcal{I}_n$, to prove the result it would be enough to investigate the update rule over two iterated applications of G_k of player i, i.e. the value $(G_k^2 a)_i$ takes. Referring back to the previous table, notice that in the case of each threshold distribution over $\{p_i, i, s_i\}$, for $(G_k^2 a)_i$ to be equal to $\mathbb{B}$, the condition $a_i = \mathbb{B}$ is either sufficient or necessary. In the case where $a_i = \mathbb{B}$ is sufficient, if $a_i = \mathbb{B}$ then $(G_k^{2m} a)_i = \mathbb{B}$ for all positive integers m, and so $\mathbb{B}$ is a strong assignment for player i. Likewise, in the case where $a_i = \mathbb{B}$ is necessary, if $a_i = \mathbb{W}$ then $(G_k^{2m} a)_i = \mathbb{W}$ for all positive integers m, and so $\mathbb{W}$ is a strong assignment for player i. $\square$

In particular, c.1, c.2, c.3 and c.5 correspond to $\mathbb{B}$ being a strong assignment for player i, and c.4, c.6, c.7 and c.8 correspond to $\mathbb{W}$ being a strong assignment for player i. We now characterize the length of the convergence cycles.

Proposition 4.1.3. *For any positive even integer n greater than 2, any 2-regular graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$ taking values in $\{1,2\}$, each cycle C in $CYCLE_n(G,k)$ has cardinality less than or equal to 2.*

Proof. Let a be an action configuration in $\mathcal{A}_n$. Suppose we construct the sequence $a, G_k a, G_k^2 a, G_k^3 a, \cdots$. For notational convenience, let us denote $G_k^T a$ by a^T. We consider the subsequence $a^0, a^2, a^4, \cdots$, choose a player i in $\mathcal{I}_n$ and then observe the evolution of the action played by player i over two time step, i.e. we consider the sequence $a_i^0, a_i^2, a_i^4, \cdots$. Without any loss of generality, let us assume that $\mathbb{B}$ is the strong assignment. Either $\mathbb{B}$ appears in the sequence or $\mathbb{B}$ does not appear in the sequence. If it does appear, then there exists a positive integer M such that $a_i^{2m} = \mathbb{B}$ for all $m \geq M$. If it does not appear, then $a_i^{2m} = \mathbb{B}$ for all non-negative integers m. Either way, for every player i, there exists a non-negative integer T_i and an action c in $\{\mathbb{W}, \mathbb{B}\}$ such that $a_i^{2m} = c$ for all $m \geq T_i$. If we set $T = \max_i T_i$, then there exists an action profile $\hat{a}$ such that $a^{2m} = \hat{a}$ for all $m \geq T$. We then get that $\{\hat{a}, G_k \hat{a}\}$ is the cycle reached from a. It follows that if we let C be any cycle in $CYCLE_n(G, k)$ and we let a be an action configuration in C, then necessarily $C = \{a, G_k a\}$. $\square$

We transition to investigate the behavior when the graph structure is complete.

4.2 On Convergence Cycles for Complete Graphs

Given some positive integer n, we consider a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is the complete graph. We put no restriction on nodes to be valid. We then define the subset $\mathcal{B}_k$ of $\mathcal{A}_n$ such that a belongs to $\mathcal{B}_k$ if and only if for every player i in $\mathcal{I}_n$, if player i plays $\mathbb{B}$ in a, then each player j with $k_j < k_i$ plays $\mathbb{B}$ in a (or equivalently, if player i plays $\mathbb{W}$ in a, then each player j with $k_j > k_i$ plays $\mathbb{W}$ in a). The reason for defining the set $\mathcal{B}_k$ is to perform the analysis with action configurations that are 'well behaved'. The following proposition states that an action configuration in $\mathcal{B}_k$ is reached from any action configuration in at most one application of G_k.

Proposition 4.2.1. *For any complete graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$, the map G_k maps every action profile in $\mathcal{A}_n$ to an action profile in $\mathcal{B}_k$, i.e. $G_k(\mathcal{A}_n) \subset \mathcal{B}_k$.*

Proof. Suppose there exists an action configuration a in $\mathcal{A}_n$ such that $G_k a$ does not belong to $\mathcal{B}_k$, then there exists a player i and j such that $k_i > k_j$, $(G_k a)_i = \mathbb{B}$ and $(G_k a)_j = \mathbb{W}$ then $|a^{-1}(\mathbb{B})| < k_j + 1$ but $|a^{-1}(\mathbb{B})| \geq k_i$, and so $k_j + 1 > k_i$ contradicting $k_i > k_j$. $\square$

Therefore, when considering the behavior only after some finite time step, we may assume that the initial action configuration belongs to $\mathcal{B}_k$. In what follows, we prepare a path for an induction to happen, we consider several cases and characterize some behavior in each.

Proposition 4.2.2. *For any complete graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$, if k takes more than one value over $\mathcal{I}_n$, then there exists a player in $\mathcal{I}_n$ that will never change color after some finite time step.*

Proof. We shall assume that all nodes are valid, otherwise there is nothing to be done. Consider the sequence $a^0, a^1, a^2, \cdots$ where a^m denotes $G_k^m a$. Let S be the set of nodes in $\mathcal{I}_n$ having the lowest threshold in the network. Consider the case where $a^T(S) = \{\mathbb{W}\}$ for some T. The action profile a^T is $\mathbb{W}$ everywhere on $\mathcal{I}_n$ since a^T belongs to $\mathcal{B}_k$, then a^{T+m} is $\mathbb{W}$ everywhere for all positive integers m and the result follows. We consider the case where $a^m(S)$ contains $\mathbb{B}$ for all m. If $a^m(S)$ contains $\mathbb{W}$ for infinitely many m, then necessarily $a^m(S) = \{\mathbb{W}, \mathbb{B}\}$ for all m greater than some M otherwise there would exist a T such that $a^T \subset_{\mathbb{B}} a^{T+1}$ where $a^{T+1}(S)$ does not contain $\mathbb{W}$, that is, all the players in S in all subsequent action profiles never play $\mathbb{W}$ again by monotonicity. It then follows since a^m is in $\mathcal{B}_k$ for all m that $a_j^m = \mathbb{W}$ for all m greater than M where j is a node having the highest threshold in the network. Finally, if $a^m(S)$ contains $\mathbb{W}$ finitely many times, then every node in S has the color $\mathbb{B}$ after some finite time step. □

Proposition 4.2.3. *For any complete graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$, if k takes only one value over $\mathcal{I}_n$ and n is odd, then all players will never flip after some finite time step.*

Proof. Let a be any action configuration in $\mathcal{A}_n$. Suppose at least two players play different actions in a, otherwise the result trivially follows. We partition $\mathcal{I}_n$ into two non-empty sets B and W such that B contains all players playing $\mathbb{B}$ in a, and W contains all players playing $\mathbb{W}$ in a. Suppose k takes the constant value b. If $b > |B|$ then all players will play $\mathbb{W}$ in $G_k a$. Similarly, if $b < |B|$, all players will play $\mathbb{B}$ in $G_k a$. Finally, for the case where $k = |B|$, all players in B will play $\mathbb{W}$ in $G_k a$ and all players in W will play $\mathbb{B}$ in $G_k a$, but since n is odd, either $|W| > |B|$ or $|W| < |B|$, and the result follows by a symmetric argument of the case where $k \neq |B|$. □

Proposition 4.2.4. *For any complete graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$ that takes only one value over $\mathcal{I}_n$. Assume $k \neq n/2$ over $\mathcal{I}_n$ and n is even. Then no player switches action after some finite time step.*

Proof. We will assume that all nodes are valid, otherwise no work is to be done. If $k \neq n/2$ then either $k < n/2$ or $k > n/2$. Let a in $\mathcal{A}_n$ be given, and define B and W

to be the set of players playing $\mathbb{B}$ and $\mathbb{W}$ respectively in a. If $k > |B|$, then $G_k a$ is equal to $\mathbb{W}$ over $\mathcal{I}_n$ and no player ever switches again. If $k < |B|$ then $G_k a$ is equal to $\mathbb{B}$ over $\mathcal{I}_n$ and no player ever switches again. Finally, if $k = |B|$ then $G_k a$ is equal to $\mathbb{B}$ on W and equal to $\mathbb{W}$ on B. In that case, $k \neq |W|$ and so the result follows by repeating the proof on $G_k a$. □

Proposition 4.2.5. *For any complete graph G in $\mathcal{G}_n$, any threshold distribution k in $\mathcal{K}_n$ and any action configuration a in $\mathcal{A}_n$, if $k = n/2$ over $\mathcal{I}_n$, n is even and $|a^{-1}(\mathbb{B})| = |a^{-1}(\mathbb{W})|$, then all players will flip actions at every time step.*

Proof. Let B and W be the set of players in $\mathcal{I}_n$ playing $\mathbb{B}$ and $\mathbb{W}$ respectively. Since $|a^{-1}(\mathbb{B})| = |a^{-1}(\mathbb{W})|$ and $k = n/2$, then $k = |a^{-1}(\mathbb{B})|$ and so all players in W will play $\mathbb{B}$ in $G_k a$ and all players in B will play $\mathbb{W}$ in $G_k a$ and $|(G_k a)^{-1}(\mathbb{B})| = |(G_k a)^{-1}(\mathbb{W})|$. □

Proposition 4.2.6. *For any complete graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$ and any action configuration a in $\mathcal{A}_n$, if $k = n/2$ over $\mathcal{I}_n$, n is even and $|a^{-1}(\mathbb{B})| \neq |a^{-1}(\mathbb{W})|$, then no player switches action after some finite time step.*

Proof. Let B and W be the set of players in $\mathcal{I}_n$ playing $\mathbb{B}$ and $\mathbb{W}$ respectively. Since $|a^{-1}(\mathbb{B})| \neq |a^{-1}(\mathbb{W})|$, then $k \neq |a^{-1}(\mathbb{B})|$. If $k < |a^{-1}(\mathbb{B})|$ then $G_k a$ will be equal to $\mathbb{W}$ over $\mathcal{I}_n$ and no node will ever switch action. Similarly, if $k > |a^{-1}(\mathbb{B})|$, then $G_k a$ will be equal to $\mathbb{B}$ over $\mathcal{I}_n$ and no node will ever switch action. □

We now characterize the lengths of the cycles.

Theorem 4.2.7. *For any positive even integer n, any complete graph G in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$, every cycle C in $CYCLE_n(G, k)$ has cardinality less than or equal to 2.*

Proof. We prove the statement by induction. The statement is trivially true for $n = 1$, and easily checked to be true for $n = 2$. So, let us suppose that the statement is true for all positive integers not greater than n, we show that it is true for $n + 1$. Let G be a complete graph in $\mathcal{G}_{n+1}$ and k a threshold distribution in $\mathcal{K}_{n+1}$. Suppose there exists a node in G that is non-valid, then after a finite time step we may apply Proposition 3.2.6 removing the node and updating the thresholds accordingly. The result would follow from the assumption that the statement holds true for n or less. If k takes only one value over $\mathcal{I}_n$, then the result follows from Propositions 4.2.3-4.2.6. If k takes more than one value of $\mathcal{I}_n$, then by Proposition 4.2.2 there exists at least one player that will never flip after some time step. Apply Proposition 3.2.6 removing the node and updating the thresholds accordingly when this node stops flipping actions. The result again follows from the assumption that the statement holds true for n or less. $\square$

We transition to study the behavior on trees.

4.3 On Convergence Cycles for Trees

We consider in this section dynamics on trees, namely acyclic connected graphs. In this section, the letter T shall always be used to denote trees, and never time as was done sometimes in previous sections. Given a tree T in $\mathcal{G}_n$, if we label a node r in $\mathcal{I}_n$ as *root*, the children of node i (with respect to the root r) are all the neighbors of i, not lying on the path from the root r to node i. Finally, a leaf (with respect to r) in the tree T is a node having degree 1. Fortunately, strong assignments appear in the dynamics on trees. We begin by stating the following proposition:

Proposition 4.3.1. *For any positive even integer n, any tree T in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$ such that all nodes are valid with respect to (G, k), pick a root r for the tree, then for every node i where all its children (with respect to r) are leaves, i has at least one strong assignment. In particular, if $k_i > 1$, then $\mathbb{W}$ is a strong assignment and if $k_i < d_i$ then $\mathbb{B}$ is a strong assignment.*

Proof. We know that for each node i where all its children (with respect to r) are leaves, node i has at least $d_i - 1$ leaves connected to it. Since all nodes are considered to be valid, then all leaves has a threshold of 1. Suppose $k_i > 1$, it then follows that

$(G_k^2 a)_i = a_i \wedge \phi(a)$, for some map ϕ from $\mathcal{A}_n$ into $\{\mathbb{W}, \mathbb{B}\}$. Suppose $k_i < d_i$, it then follows that $(G_k^2 a)_i = a_i \vee \phi(a)$, for some map ϕ from $\mathcal{A}_n$ into $\{\mathbb{W}, \mathbb{B}\}$. □

In this case, note that if $1 < k_i < d_i$, then i has both $\mathbb{B}$ and $\mathbb{W}$ as strong assignment. This fact implies that i will never change its color over two time steps. It is to note that the proposition considers only the case where all nodes in concern are valid.

Aside being acyclic, trees enjoy bipartiteness: a crucial property that will be heavily relied on when considering general graphs. We begin to convey how the bipartite property of graphs may be exploited. The definitions and results to follow apply to general bipartite graphs.

Definition 4.3.2. *Let P be a subset of $\mathcal{I}_n$, for (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$, we define $G_k|_P$ to be the restriction of G_k to act on the actions of the players in P. Formally, for a in $\mathcal{A}_n$,*

$$(G_k|_P a)_i = \begin{cases} (G_k a)_i & \text{if } i \in P \\ a_i & \text{if } i \notin P \end{cases}$$

It is to note that we are not restricting the domain of the function, $G_k|_P$ is indeed a map from $\mathcal{A}_n$ into $\mathcal{A}_n$. To proceed, it is known that any bipartite graph has a 2-(node)-coloring, we avoid the wording *coloring* to avoid confusion. Instead, we define *2-Partitions.* Let $\mathcal{G}_n^b$ be the set of all connected undirected bipartite graphs defined over the vertex set $\mathcal{I}_n$.

Definition 4.3.3. *Given a graph $G(\mathcal{I}_n, E^b)$ in $\mathcal{G}_n^b$, a 2-Partition of $\mathcal{I}_n$ with respect to G, is a pair (P_o, P_e) of disjoint subsets of $\mathcal{I}_n$ such that $P_o \cup P_e = \mathcal{I}_n$ and there does not exist an (i, j) in $P_o^2 \cup P_e^2$ such that $ij \in E^b$.*

We eventually restrict G_k to act on the nodes in P_o and P_e separately. For convention, o would refer to *odd* and e to *even.* The dynamics will be presented in such a way, that nodes in P_o (resp. P_e) will be allowed to change actions only at odd (resp. even) time steps. Let us first clearly define a partition of a set.

Definition 4.3.4. *Let X be a set, a partition $P_1, \cdots, P_m$ of X is a finite collection of disjoint non-empty subsets of X whose union is X.*

The definition to follow serves mainly as a notational clarification, its technical value is rather intuitive.

Definition 4.3.5. *Consider a function f mapping $\mathcal{I}_n$ into some set. Let $P_1, \cdots, P_m$ be a partition of $\mathcal{I}_n$, and let $f\lceil P_l$ be the restriction of f to have domain P_l. Let π be any permutation on $\{1, \cdots, m\}$, we consider f to be equal to $(f\lceil P_{\pi(1)}, \cdots, f\lceil P_{\pi(m)})$.*

Given a 2-Partition, we may 'decouple' the dynamics and the following identities would emerge:

Proposition 4.3.6. *Given a pair (G, k) in $G_n^b \times \mathcal{K}_n$, if we consider a 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to G, then:*

1. $G_k a = ((G_k|_{P_o} a)\lceil P_o, (G_k|_{P_e} a)\lceil P_e)$
2. $G_k|_{P_e} G_k|_{P_o} a = (G_k a\lceil P_o, G_k^2 a\lceil P_e)$
3. $G_k^2 a = ((G_k|_{P_e} G_k|_{P_o} a)\lceil P_e, (G_k|_{P_o} G_k|_{P_e} a)\lceil P_o)$.

Proof. The fact that $G_k a = ((G_k|_{P_o} a)\lceil P_o, (G_k|_{P_e} a)\lceil P_e)$ follows from the definition of G_k. We have $(G_k|_{P_o} a)\lceil P_o = (G_k a)\lceil P_o$, so $(G_k|_{P_e} G_k|_{P_o} a)\lceil P_o = (G_k a)\lceil P_o$, and on the other hand since $G_k|_{P_e}$ modifies the action of the players in P_e based only on the actions in P_o, we get:

$$\begin{aligned}
G_k|_{P_e} G_k|_{P_o} a\lceil P_e &= G_k|_{P_e}((G_k a)\lceil P_o, a\lceil P_e)\lceil P_e \\
&= G_k|_{P_e}((G_k a)\lceil P_o, (G_k a)\lceil P_e)\lceil P_e \\
&= G_k|_{P_e} G_k a\lceil P_e \\
&= G_k^2 a\lceil P_e.
\end{aligned}$$

As for the last statement,

$$\begin{aligned}
G_k^2 a\lceil P_o &= G_k(G_k a\lceil P_e, G_k a\lceil P_o)\lceil P_o \\
&= G_k|_{P_o}(G_k a\lceil P_e, G_k a\lceil P_o)\lceil P_o \\
&= G_k|_{P_o}(G_k|_{P_e} a\lceil P_e, G_k|_{P_e} a\lceil P_o)\lceil P_o \\
&= G_k|_{P_o} G_k|_{P_e} a\lceil P_o.
\end{aligned}$$

Similarly, we get $G_k^2 a \restriction P_e = G_k|_{P_e} G_k|_{P_o} a \restriction P_o$. $\square$

This fact allows us so say something about the sequence $a, G_k a, G_k^2 a, \cdots$ for all a by investigating instead $b, G_k|_{P_e} G_k|_{P_o} b, (G_k|_{P_e} G_k|_{P_o})^2 b, \cdots$ for all b. In some way, the proposition allows us to observe the process diagonally, in a zig-zag fashion. We shall perform that, and we state the following lemma to formalize the idea.

Proposition 4.3.7. *Given a pair (G, k) in $G_n^b \times \mathcal{K}_n$, for every cycle C in $CYCLE_n(G, k)$, if for some action configuration a in C, for any 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to G we have $(G_k|_{P_e} G_k|_{P_o} a) \restriction P_e = a \restriction P_e$ then C has a cardinality of at most 2.*

Of course, the condition could have also been written as $(G_k|_{P_o} G_k|_{P_e} a) \restriction P_o = a \restriction P_o$, but that statement is equivalent to the statement in the proposition because it holds for any 2-Partition.

Proof. From Proposition 4.3.6, we have $G_k^2 a = ((G_k|_{P_e} G_k|_{P_o} a) \restriction P_e, (G_k|_{P_o} G_k|_{P_e} a) \restriction P_o)$. Let C be a cycle in $CYCLE_n(G, k)$, if we have $(G_k|_{P_e} G_k|_{P_o} a) \restriction P_e = a \restriction P_e$ for some action configuration in C, then $G_k^2 a = a$ and so C has a cardinality of at most 2. $\square$

To prove that all cycle have cardinality at most 2, we will study for all 2-Partitions (P_o, P_e) and every a in $\mathcal{A}_n$, the sequence $a, G_k|_{P_e} G_k|_{P_o} a, (G_k|_{P_e} G_k|_{P_o})^2 a$ and show that this sequence is eventually constant, i.e. there exists a finite time step where all terms become equal. If that is the case, then for any a in $\mathcal{A}_n$ the sequence $a, G_k^2 a, G_k^4 a, \cdots$ is eventually constant and so cycles cannot have a cardinality greater than 2. To this end, we note the following fact:

Proposition 4.3.8. *Given a pair (G, k) in $\mathcal{G}_n^b \times \mathcal{K}_n$ and a 2-Partition (P_o, P_e) of $\mathcal{I}_n$, for every player i in $\mathcal{I}_n$, if $c \in \{\mathbb{W}, \mathbb{B}\}$ is a strong assignment for player i, then for every a in $\mathcal{A}_n$, if $(G_k|_{P_e} G_k|_{P_o} a)_i = c$, then $((G_k|_{P_e} G_k|_{P_o})^m a)_i = c$ for all positive integers m.*

Proof. If i belongs to P_e, we have $(G_k|_{P_e} G_k|_{P_o} a)_i = (G_k^2 a)_i$, and the result then follows from the definition of strong assignment. If i belongs to P_o, then $(G_k|_{P_e} G_k|_{P_o} a)_i =$

$(G_k|_{P_o}a)_i$, and so if $(G_k|_{P_e}G_k|_{P_o}a)_i = c$ then $(G_k|_{P_o}a)_i = c$ and since:

$$\begin{aligned}((G_k|_{P_e}G_k|_{P_o})^m a)_i &= (G_k|_{P_e}(G_k|_{P_o}G_k|_{P_e})^{m-1}G_k|_{P_o}a)_i \\ &= ((G_k|_{P_o}G_k|_{P_e})^{m-1}G_k|_{P_o}a)_i \\ &= (G_k^{2m-2}G_k|_{P_o}a)_i\end{aligned}$$

The result would follow from the definition of strong assignment. □

We now specialize Proposition 3.2.7 to meet our needs for the bipartite case.

Proposition 4.3.9. *Let (G, k) be a pair in $\mathcal{G}_n^b \times \mathcal{K}_n$, (P_o, P_e) be a 2-Partition of $\mathcal{I}_n$ and a an action configuration in $\mathcal{A}_n$. If there exists a player i and an action c, such that $((G_k|_{P_e}G_k|_{P_o})^m a)_i = c$ for all non-negative integers m, consider H to be the induced subgraph of G over $\mathcal{I}_n \backslash \{i\}$. Suppose G' is a connected component of H with vertex set $\mathcal{J}$, define a' to be the action configuration a restricted to the players in $\mathcal{J}$, P'_o and P'_e te be $P_o \cap \mathcal{J}$ and $P_e \cap \mathcal{J}$ respectively, and define k' to be the map from $\mathcal{J}$ into $\mathbb{N}$ such that $k' = k$ on $\mathcal{J} \backslash \mathcal{N}_i$, $k' = k$ on $\mathcal{N}_i \cap \mathcal{J}$ if $c = \mathbb{W}$ and $k' = (k-1) \vee 0$ on $\mathcal{N}_i \cap \mathcal{J}$ if $c = \mathbb{B}$. Then,*

$$((G_k|_{P_e}G_k|_{P_o})^m a)_j = ((G'_{k'}|_{P'_e}G'_{k'}|_{P'_o})^m a')_j,$$

for all non-negative integers m and all players j in $\mathcal{J}$.

Proof. It would be enough to show that the local decision rules of the players in $\mathcal{J} \cap \mathcal{N}_i$ does not change, simply that

$$(G_k|_{P_e}G_k|_{P_o}a)_j = (G'_{k'}|_{P'_e}G'_{k'}|_{P'_o}a')_j.$$

Let j be a player in $\mathcal{J} \cap \mathcal{N}_i$, either j belongs to P_o or j belongs to P_e. Suppose j belongs to P_o, then $(G_k|_{P_e}G_k|_{P_o}a)_j = (G_k|_{P_o}a)_j$, and $(G_k|_{P_o}a)_j = \mathbb{B}$ if and only if at least k_j nodes in $\mathcal{N}_j$ play $\mathbb{B}$ in a, or equivalently at least k'_j nodes in $\mathcal{N}_j \backslash \{i\}$ play $\mathbb{B}$ in a', since k' takes into account the action of player i. And that is equivalent to $(G'_{k'}|_{P'_e}G'_{k'}|_{P'_o}a')_j = \mathbb{B}$. Similarly, if j belongs to P_e we get that $(G_k|_{P_e}G_k|_{P_o}a)_j = (G_k|_{P_e}(G'_{k'}|_{P'_o}a', c\restriction\{i\})_j = G'_{k'}|_{P'_e}G'_{k'}|_{P'_o}a$. $\square$

With this settled, we go now to proving our theorem.

Proposition 4.3.10. *For any positive integer n, any tree T in $\mathcal{G}_n$ and any threshold distribution k in $\mathcal{K}_n$, every cycle C in $CYCLE_n(T, k)$ has cardinality less than or equal to 2.*

Proof. We prove this statement by induction. The statement can be checked for all one node, two node networks. So let us suppose that it holds for all graphs with size less than or equal to n where $n \geq 3$. Given a tree of size $n+1$, if there exists a non-valid node, then after some finite time-step those nodes never flip, apply Proposition **??** removing those node and updating the thresholds accordingly, the result follows since it holds on the connected components obtained. If all nodes are valid, since $n \geq 2$ necessarily there exists a parent and a leaf, by the previous lemma it has a strong assignment, and so it stops flipping after some finite time step. When it stops flipping, we now apply the Proposition **??**, removing that node and updating the thresholds accordingly, the result holds since it holds on the connected component. $\square$

We proceed to extend such results to general graphs. Unfortunately, the notion of strong assignment does not extend to general case, e.g. consider the complete graph considered earlier. We construct a different approach.

4.4 On Convergence Cycles for General Graphs

Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ we will have this pair undergo two procedures: a bipartite expansion and a symmetric-expansion. We introduced them in what follows.

Recall that we denote by $\mathcal{G}^b_n$ the set of all connected undirected bipartite graphs defined over the vertex set $\mathcal{I}_n$.

Definition 4.4.1. *Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is non-bipartite, we construct a pair (G', k') in $\mathcal{G}^b_{2n} \times \mathcal{K}_{2n}$ as outlined in procedure to follow. We refer to (G', k') as the bipartite-expansion of (G, k).*

Bipartite-expansion procedure — Given a pair (G,k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is non-bipartite and has edge set E, we construct a pair (G',k') in $\mathcal{G}^b_{2n} \times \mathcal{K}_{2n}$ as follows. We suppose G' is equal to $(\mathcal{I}_{2n}, E')$, and partition $\mathcal{I}_{2n}$ into two sets $\mathcal{I}_n$ and J_n. We define a bijection ϕ from $\mathcal{J}_n$ into $\mathcal{I}_n$ and define E' to be the set of undirected edges on $\mathcal{I}_{2n}$ such that for i and j in $\mathcal{I}_{2n}$, $\{i,j\} \in E'$ if and only if $(i,j) \in \mathcal{I}_n \times \mathcal{J}_n$ and $\{i, \phi(j)\} \in E$. Finally, set k' to be equal to k on $\mathcal{I}_n$ and $k \circ \phi$ on $\mathcal{J}_n$.

We define $\mathcal{S}_n$ to be the set of all pairs (G,k) in $\mathcal{G}_n \times \mathcal{K}_n$ such for each player i in $\mathcal{I}_n$, the degree d_i is odd and k_i is equal to $(d_i+1)/2$. We refer to $\mathcal{S}_n$ as the set of *symmetric models*, in the sense that for (G,k) in $\mathcal{S}_n$ the property is such that for any action profile a in $\mathcal{A}_n$, and any player i, the action $(G_k a)_i$ is the action played by the majority in $\mathcal{N}_i$ with respect to the action profile a. In this case, the two actions $\mathbb{B}$ and $\mathbb{W}$ are treated as having equal weights by all players in the network.

Definition 4.4.2. *Given a pair (G,k) in $(\mathcal{G}_n \times \mathcal{K}_n)\backslash\mathcal{S}_n$, we construct a pair (G',k') in $\mathcal{G}'_n \times \mathcal{K}'_n$ as outline by procedure to follow. We refer to (G',k') as a one-step symmetric-expansion of (G,k).*

One-step symmetric-expansion procedure — Given a pair (G,k) in $(\mathcal{G}_n \times \mathcal{K}_n)\backslash\mathcal{S}_n$, we construct a pair (G',k') in $\mathcal{G}'_n \times \mathcal{K}'_n$ as follows. We suppose that G is equal to $(\mathcal{I}_n, E)$, and choose a player i in $\mathcal{I}_n$ such that either d_i is even, or d_i is odd and k_i is not equal to $(d_i+1)/2$. Surely such a node exists since (G,k) does not belong to $\mathcal{S}_n$. We call the node i the **pivot** node in the one-step symmetric-expansion of (G,k) into (G',k'). Let b_i be an integer equal to k_i, and consider w_i an integer equal to $d_i - b_i + 1$. In this sense, if a is an action configuration in $\mathcal{A}_n$, b_i would be considered to be the least number of $\mathbb{B}$-playing neighbors needed by player i to play $\mathbb{B}$ when G_k acts on a, whereas w_i would be the least number of $\mathbb{W}$-playing neighbors needed by player i to play $\mathbb{W}$. We shall construct an instance (G',k') in $\mathcal{G}_{n+3b_i+3w_i} \times \mathcal{K}_{n+3b_i+3w_i}$. We suppose that G' is equal to $(\mathcal{I}_{n+3b_i+3w_i}, E')$ and partition $\mathcal{I}_{n+3b_i+3w_i}$ into $\mathcal{I}_n, P^w_1, \cdots, P^w_{b_i}, P^b_1, \cdots, P^b_{w_i}$ where each partition different than $\mathcal{I}_n$ has cardinality exactly equal 3.

We define E' to be the undirected set of edges such that E' contains E. Furthermore, for every m, suppose $P_m^w = \{j, j', j''\}$, we let E' contain jj', jj'' and ij. Similarly, for every l, suppose $P_l^b = \{j, j', j''\}$ we let E' contain jj', jj'' and ij. To visualize the obtained graph structure G', we attached $b_i + w_i$ 3-node Y-shaped graphs to node i.

Finally, we set k' to be equal to k on $\mathcal{I}_n \backslash \{i\}$, to be equal to $(d_i + b_i + w_i)/2$ at i, equal to 2 on the remaining nodes having degree 3 and equal to 1 everywhere else.

Iterated one-step symmetric-expansion of pair in $(\mathcal{G}_n \times \mathcal{K}_n) \backslash \mathcal{S}_n$ eventually yields a pair in $\mathcal{S}_n$. We formalize the idea:

Definition 4.4.3. *Given a pair (G_0, k_0) in $(\mathcal{G}_n \times \mathcal{K}_n) \backslash \mathcal{S}_n$, we construct a finite sequence $(G_0, k_0), (G_1, k_1), \cdots, (G_m, k_m)$ for some positive integer m, where (G_l, k_l) is a one-step symmetric-expansion of (G_{l-1}, k_{l-1}) and (G_m, k_m) belongs to $\mathcal{S}_{n'}$ for some n'. We refer to (G_m, k_m) as the symmetric-expansion of (G, k).*

A specific note to be made, is that the symmetric-expansion of (G, k) is uniquely defined (up to isomorphism[3]) regardless of the order the pivot nodes were chosen. However, we shall not write a formal proof for this fact, it would suffice to say that when we perform the one-step symmetric expansion of a pair (G, k), we leave all neighborhoods and thresholds unchanged for all nodes different than the pivot.

We proceed to present some properties of the (one-step) symmetric-expansion and bipartite-expansion, and begin linking expansions to convergence cycle properties. Some of the claims are rather and simple, nevertheless we mention them first for completeness and second to develop intuition and ease while dealing with such matters, and ensure that no insight escapes.

Proposition 4.4.4. *Given a pair (G, k) in $(\mathcal{G}_n \times \mathcal{K}_n) \backslash \mathcal{S}_n$. Let (G', k') be the one-step symmetric-expansion of (G, k), then G is bipartite if and only if G' is biparite.*

Proof. If G' is biparite, then G is bipartite being an induced subgraph of G'. To show the converse, let i be the pivot node in the one-step symmetric-expansion of (G, k)

[3] Two pairs (G, k) and (G', k') in $\mathcal{G}_n \times \mathcal{K}_n$ are said to be isomorphic if there exists a bijective map ϕ from $V(G)$ into $V(G')$ such that $ij \in E(G)$ if and only if $\phi_i \phi_j \in E(G')$ and $k_i = (k' \circ \phi)_i$ for i in $V(G)$.

into (G', k'), and let (P_o, P_e) be a two partition of $\mathcal{I}_n$ with respect to G. Let R and L be the set of nodes in $V(G')\backslash\mathcal{I}_n$ having degree equal to 3 and 1 respectively. If i belongs to P_o, then $(P_o \cup L, P_e \cup R)$ is a 2-partition of $V(G')$ with respect to G'. Therefore G' is bipartite. $\square$

Proposition 4.4.5. *Given a pair (G, k) in $(\mathcal{G}_n\backslash\mathcal{G}_n^b)\times\mathcal{K}_n$. Let (G', k') be the bipartite-expansion of (G, k), then (G, k) belongs to $\mathcal{S}_n$ if and only if (G', k') belongs to $\mathcal{S}_{2n}$.*

Proof. If (G, k) belongs to $\mathcal{S}_n$, then clearly (G', k') belongs to $\mathcal{S}_{2n}$ by construction. Conversely, if (G', k') belongs to $\mathcal{S}_{2n}$, then $k = k' \restriction \mathcal{I}_n$ and $d_i^G = d_i^{G'}$ for i in $\mathcal{I}_n$. $\square$

Definition 4.4.6. *We define the set $\mathcal{M}$ to be a subset $\bigcup_{n\geq 1} \mathcal{G}_n\times\mathcal{K}_n$ such that (G, k) in $\mathcal{G}_n\times\mathcal{K}_n$ belongs to $\mathcal{M}$ if and only if for every C in $CYCLE_n(G, k)$, the cardinality of C is less than or equal to 2.*

What we ultimately show is that $\mathcal{M} = \bigcup_{n\geq 1} \mathcal{G}_n\times\mathcal{K}_n$.

Lemma 4.4.7. *Given a pair (G, k) in $(\mathcal{G}_n \times \mathcal{K}_n)\backslash\mathcal{S}_n$, define (G', k') to be the one-step symmetric-expansion of (G, k). If (G', k') belongs to $\mathcal{M}$ then (G, k) belongs to $\mathcal{M}$.*

Proof. Let node i be the *pivot* node in the one-step symmetric-expansion of (G, k) into (G', k'). We suppose G' belongs to $\mathcal{G}_{n'}$, and consider the induced subgraph H in G' over the vertex set $\mathcal{I}_{n'}\backslash\mathcal{I}_n$. The subgraph H necessarily consists (by construction of G') of $d_i + 1$ connected components, each consisting of three vertices. We set integers b_i and w_i to be equal to k_i and $d_i - k_i + 1$ respectively, we then partition $\mathcal{I}_{n'}\backslash\mathcal{I}_n$ into $W_1, \cdots, W_{b_i}, B_1, \cdots, B_{w_i}$ such that each partition contains the set of nodes in one of the connected components. Suppose that (G', k') belongs to $\mathcal{M}$, we show that (G, k) belongs to $\mathcal{M}$. Let C be an element of $CYCLE_n(G, k)$ and suppose C has cardinality greater or equal to 3. In particular, suppose C is equal to $\{a_1, \cdots, a_m\}$ for $m \geq 3$ where $a_{l+1} = G_k a_l$ for $1 \leq l < m$ and $a_1 = G_k a_m$.

We define a map α from $\mathcal{A}_n$ into $\mathcal{A}_{n'}$ such that for a in $\mathcal{A}_n$,

$$\alpha a = \begin{cases} a & \text{on } \mathcal{I}_n \\ \mathbb{B} & \text{on } B_1 \cup \cdots \cup B_{w_i} \\ \mathbb{W} & \text{on } W_1 \cup \cdots \cup W_{b_i} \end{cases}.$$

First, the map α is clearly injective. Then, $\alpha a_1, \cdots, \alpha a_m$ are distinct elements of $\mathcal{A}_{n'}$. Second, it can be checked that

$$\alpha(G_k a) = G'_{k'}(\alpha a).$$

To see this fact, notice that for every b_1 and b_2 in $\mathcal{A}_n$ every node in $\mathcal{I}_{n'} \backslash \mathcal{I}_n$ has the same color both in αb_1 and αb_2, and the same color both in αb_1 and $G'_{k'}\alpha b_1$. Since every node in $\mathcal{I}_n$ other than the pivot i keeps the same neighborhood and threshold, we need only show that $(\alpha G_k a)_i = (G'_{k'}\alpha a)_i$. To this end, node i is $\mathbb{B}$ in $G_k a$ if and only if at least k_i neighbors of i in G are $\mathbb{B}$ in a, and that is the case if and only if at least $k_i + w_i$ neighbors of i in G' are $\mathbb{B}$ in αa, or equivalently at least $d^G(i) + 1 = (d^{G'}(i) + 1)/2$ neighbors of i in G' are $\mathbb{B}$ in αa. Finally, it follows that $\alpha C = \{\alpha a_1, \cdots, \alpha a_m\}$ is a cycle in $CYCLE_{n'}(G', k')$ contradicting the fact that $m \geq 3$. $\square$

Lemma 4.4.8. *Given a pair (G, k) in $(\mathcal{G}_n \times \mathcal{K}_n) \backslash \mathcal{S}_n$, define (G', k') to be the symmetric-expansion of (G, k). If (G', k') belongs to $\mathcal{M}$ then (G, k) belongs to $\mathcal{M}$.*

Proof. Given that the pair (G, k) belongs to $(\mathcal{G}_n \times \mathcal{K}_n) \backslash \mathcal{S}_n$, we can construct a finite sequence $(G_1, k_1), \cdots, (G_{m-1}, k_{m-1})$ in such a way that (G_1, k_1) is the one-step symmetric expansion of (G, k), (G', k') is the one-step symmetric-expansion of (G_{m-1}, k_{m-1}) and (G_l, k_l) is the one-step symmetric-expansion of (G_{l-1}, k_{l-1}) for $1 < l < m$. If (G', k') belongs to $\mathcal{M}$, then (G_{m-1}, k_{m-1}) belongs to $\mathcal{M}$ by Lemma 4.4.7. Recursively, it follows that (G, k) belongs to $\mathcal{M}$. $\square$

Lemma 4.4.9. *Given a pair (G, k) in $(\mathcal{G}_n \backslash \mathcal{G}_n^b) \times \mathcal{K}_n$, define (G', k') to be the bipartite-expansion of (G, k). If (G', k') belongs to $\mathcal{M}$ then (G, k) belongs to $\mathcal{M}$.*

Proof. The graph G' is bipartite and has vertex set $\mathcal{I}_{2n}$, let us partition $\mathcal{I}_{2n}$ into $\mathcal{I}_n$ and $\mathcal{J}$, then $(\mathcal{I}_n, \mathcal{J})$ forms a 2-Partition with respect to G' by construction. We define a bijection ϕ from $\mathcal{J}$ into $\mathcal{I}_n$ such that for j_1 and j_2 in $\mathcal{J}$, $j_1\phi(j_2) \in E'$ if and only if $\phi(j_1)\phi(j_2) \in E$. Given an action configuration in $\mathcal{A}_n$, we define the map α from $\mathcal{A}_n$ into $\mathcal{A}_{2n}$ such that for a in $\mathcal{A}_n$ we have:

$$\alpha a = (a, a \circ \phi).$$

It then follows that:

$$G'_{k'}(\alpha a) = \alpha(G_k a).$$

The map α is clearly injective, it then follows that for any cycle $C = \{a_1, \cdots, a_m\}$ in $CYCLE_n(G,k)$ with $m > 2$, $\alpha C = \{\alpha a_1, \cdots, \alpha a_m\}$ is a cycle in $CYCLE_n(G', k')$ contradicting the fact that (G', k') belongs to $\mathcal{M}$. $\square$

Proposition 4.4.10. *Given a pair (G,k) in $((\mathcal{G}_n \backslash \mathcal{G}_n^b) \times \mathcal{K}_n) \backslash \mathcal{S}_n$, we define (G_1, k_1) and (G_2, k_2) to be respectively the bipartite-expansion and the symmetric expansion of (G,k). If (G'_1, k'_1) is the symmetric-expansion of (G_1, k_1) and (G'_2, k'_2) is the bipartite-expansion of (G_2, k_2), then (G'_1, k'_1) is equal to (G'_2, k'_2) (up to isomorphism[4]).*

Proof. It suffices to prove existence of a bijective map ϕ from $V(G'_2)$ to $V(G'_1)$, such that $ij \in E(G'_1)$ if and only if $\phi_i \phi_j \in E(G'_2)$ and $(k'_1)_i = (k'_2 \circ \phi)_i$ for i in $V(G'_1)$. Such a map could be easily constructed following the expansion procedure. We omit the construction. $\square$

Lemma 4.4.11. *Given a pair (G,k) in $((\mathcal{G}_n \backslash \mathcal{G}_n^b) \times \mathcal{K}_n) \backslash \mathcal{S}_n$, define (G', k') to be the bipartite-expansion of (G,k) and (G'', k'') to be the symmetric-expansion of (G', k'). If (G'', k'') belongs to $\mathcal{M}$ then (G,k) belongs to $\mathcal{M}$.*

Proof. If (G'', k'') belongs to $\mathcal{M}$, then (G', k') belongs to $\mathcal{M}$ by Lemma 4.4.8. It follows that (G,k) belongs to $\mathcal{M}$ by Lemma 4.4.9. $\square$

Lemma 4.4.12. *The set $\mathcal{M}$ is equal to $\bigcup_{n \geq 1} \mathcal{G}_n \times \mathcal{K}_n$ if and only if $\mathcal{M}$ contains $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n)$ for every positive integer n.*

Proof. It is clear that if $\mathcal{M} = \bigcup_{n \geq 1} \mathcal{G}_n \times \mathcal{K}_n$ then $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n) \subset \mathcal{M}$. To prove the converse, given (G,k) an element of $\bigcup_{n \geq 1} \mathcal{G}_n \times \mathcal{K}_n$, we define (G', k') to be the bipartite-expansion of (G,k) and (G'', k'') to be the symmetric-expansion of (G', k'). We have that (G'', k'') belongs to $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n)$. If $\mathcal{M}$ contains $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n)$, it then follows by Lemma 4.4.11 that $\mathcal{M}$ contains (G,k). $\square$

We proceed to show that $\mathcal{M}$ contains $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n)$ for every positive integer n.

[4]See footnote 3.

Lemma 4.4.13. *Given a pair* (G, k) *in* $G_n^b \times \mathcal{K}_n$*, we consider a 2-Partition* (P_o, P_e) *of* $\mathcal{I}_n$ *with respect to* G*. Then* (G, k) *belongs to* $\mathcal{M}$ *if and only if for every* a *in* $\mathcal{A}_n$*, there exists some integer* T*, such that* $(G_k|_{P_e} G_k|_{P_o})^T a = (G_k|_{P_e} G_k|_{P_o})^{T+1} a$*.*

Proof. It would be enough to say that the statement of this lemma is equivalent to the statement of Proposition 4.3.7. □

Definition 4.4.14 (Conflict Link). *Given* (G, a) *in* $\mathcal{G}_n \times \mathcal{A}_n$ *with* $G = (\mathcal{I}_n, E)$*, we call a conflict link in* G *with respect to* a*, an element* ij *of* E *such that* a_i *and* a_j *are not equal. We denote by* $E_c^G(a)$ *the set of all conflict links in* G *with respect to* a*.*

Lemma 4.4.15. *The set* $\mathcal{M}$ *contains* $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n)$ *for every positive integer* n*.*

Proof. Let (G, k) in $\mathcal{S}_n \cap (G_n^b \times \mathcal{K}_n)$ be given and consider a 2-Partition (P_o, P_e) of $\mathcal{I}_n$. Let a be an action profile in $\mathcal{A}_n$. By Lemma 4.4.13 it would be enough to show that $(G_k|_{P_e} G_k|_{P_o})^T a = (G_k|_{P_e} G_k|_{P_o})^{T+1} a$ for some non-negative integer T. In that case, it would be enough to prove that for every b in $\mathcal{A}_n$:

$$G_k|_{P_o} b \neq b \quad \text{if and only if} \quad |E_c(G_k|_{P_o} b)| < |E_c(b)|.$$

and similarly:

$$G_k|_{P_e} b \neq b \quad \text{if and only if} \quad |E_c(G_k|_{P_e} b)| < |E_c(b)|.$$

To show that we state the following, for node i in $\mathcal{I}_n$, $a_i \neq (G_k a)_i$ if and only if the majority of the players in $\mathcal{N}_i$ are not playing a_i, or equivalently, if and only if i can decrease the number of conflict edges by switching action. □

Theorem 4.4.16. *For every positive integer* n*, every* (G, k) *in* $\mathcal{G}_n \times \mathcal{K}_n$ *and every* C *in* $CYCLE_n(G, k)$*, the cardinality of* C *is less than or equal to* 2*.*

Proof. The statement of the theorem is equivalent to $\mathcal{M} = \bigcup_{n \geq 1} \mathcal{G}_n \times \mathcal{K}_n$. The fact that $\mathcal{M} = \bigcup_{n \geq 1} \mathcal{G}_n \times \mathcal{K}_n$ follows directly from Lemma 4.4.15 and Lemma 4.4.12. □

Finally, we extend the lengths of convergence cycle results to the extension model allowing non-equal weights on edges.

4.5 On Convergence Cycles under Weighted Edges

Given a pair (G, q) in $\mathcal{G}_n \times \mathcal{Q}_n$, where G has edge set E, we assign every edge ij in E a positive weight $w_{ij} = w_{ji}$. Modifying the utility function to incorporate a weighted sum of the payoffs from neighboring interactions, we have the following rule. We denote by $a_{i,T}$ the action played by player i at time T. If we let $\underline{a}$ be the initial action configuration, namely $a_{i,0} = \underline{a}_i$, then for every positive integer T, player i plays action $a_{i,T} = \mathbb{B}$ at time step T if and only if

$$\sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{1}_{\{\mathbb{B}\}}(a_{j,T-1}) > \theta_i,$$

where we define $\theta_i = q_i \sum_{j \in \mathcal{N}_i} w_{ij}$.

We shall then expand $\mathcal{G}_n$ to incorporate the weights. We shall define $\mathcal{W}_n$ to be the set of connected graph over the vertex set $\mathcal{I}_n$ with weighted edges. Given a weighted graph W of $\mathcal{W}_n$, we shall denote the set of edges by $E(W)$, the weight on edge ij in $E(W)$ is then denoted by w_{ij}. As a natural extension of previous definitions, given a pair (W, q) we define the map W_q from $\mathcal{A}_n$ into $\mathcal{A}_n$ such that $(W_q a)_i = \mathbb{B}$ if and only if $\sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{1}_{\{\mathbb{B}\}}(a_j) \geq q_i \sum_{j \in \mathcal{N}_i} w_{ij}$.

We put no restrictions on the weights in $\mathcal{W}_n$ other than being non-negative reals. We first show that with no loss of generality, we may consider the weights to be integers.

Lemma 4.5.1. *For every pair (W, q) in $\mathcal{W}_n \times \mathcal{Q}_n$, there exists a pair (W', q') in $\mathcal{W}_n \times \mathcal{Q}_n$ where the weights are positive integers such that for every action configuration a in $\mathcal{A}_n$, we have $W_q a = W'_{q'} a$.*

Proof. It would be enough to show, that there exists a pair (W', q') in $\mathcal{W}_n \times \mathcal{Q}_n$ with rational coefficient. If that is the case, since the graph is finite, we can make all weights integral by multiplying the rational weight by some integer. This said, we state again that it should be enough to prove that given (W, q) in $\mathcal{W}_n \times \mathcal{Q}_n$ and an edge e in $E(W)$, there exists a pair (W', q') in $\mathcal{W}_n \times \mathcal{Q}_n$ such that $W_q a = W'_{q'} a$ for all a in $\mathcal{A}_n$, w'_e is rational, and $w_{e'} = w'_{e'}$ for all edges e' in $E(W) = E(W')$ different than e.

To prove this, pick an edge ij in $E(W)$, and suppose w_{ij} is irrational. Let us denote by θ_i the quantity $q_i \sum_{j\in\mathcal{N}_i} w_{ij}$, we then define the map Δ_i from $\mathcal{A}_n$ into $\mathbb{R}$ such that for a in $\mathcal{A}_n$:

$$\Delta_i(a) = \theta_i - \sum_{l\in\mathcal{N}_i\backslash\{j\}} w_{il}\mathbf{1}_{\{\mathbb{B}\}}(a_l) - w_{ij}.$$

We set:

$$\delta_i = \min_{a\in\mathcal{A}_n \,:\, \Delta_i(a)>0} |\Delta(a)|.$$

Similarly, we define the map Δ_j from $\mathcal{A}_n$ into $\mathbb{R}$ such that for a in $\mathcal{A}_n$:

$$\Delta_j(a) = \theta_j - \sum_{l\in\mathcal{N}_j\backslash\{i\}} w_{jl}\mathbf{1}_{\{\mathbb{B}\}}(a_l) - w_{ij}.$$

We also set:

$$\delta_j = \min_{a\in\mathcal{A}_n \,:\, \Delta_j(a)>0} |\Delta(a)|.$$

Both δ_i and δ_j are well defined being smallest elements in finite non-empty sets. Pick any rational weight $w_i'j$ in $[w_{ij}, w_{ij} + \delta_i \wedge \delta_j)$. Set q' to be equal to q on $\mathcal{I}_n\backslash\{i,j\}$, $q_l' = \theta_l/(\theta_l - w_{ij} + w_0)$ for $l \in \{i,j\}$. We get pair (W', q') with $w_e' = w_e$ for all $e \in E(W)$ and w_{ij} rational. $\square$

With this result in mind, we redefine the set $\mathcal{W}_n$ to be the set of connected graph over the vertex set $\mathcal{I}_n$ with integral weighted edges, i.e. weights in $\mathbb{N}$. Following a similar reason as in Chapter 3, without any loss in generality, we substitute the set Q_n with the set $\mathcal{K}_n$ as defined. For a pair (W,k) in $\mathcal{W}_n \times \mathcal{K}_n$, the map W_k extends naturally from W_q for the pair (W,q) in $\mathcal{W}_n \times \mathcal{Q}_n$, in the way that for an action configuration a in $\mathcal{A}_n$, $(W_k a)_i = \mathbb{B}$ if and only if $\sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{1}_{\{\mathbb{B}\}}(a_j) \geq k_i$. Similarly, we define $CYCLE_n(W,k)$ to extends naturally from Chapter 3.

Theorem 4.5.2. *For any positive integer n, and any pair (W,k) in $\mathcal{W}_n \times \mathcal{K}_n$, every cycle C in $CYCLE_n(W,k)$ has cardinality less than or equal to 2.*

Proof. Given a pair (W,k) in $\mathcal{W}_n \times \mathcal{K}_n$, we shall construct an instance (G,k) in $\mathcal{G}_n \times \mathcal{K}_n$. We define N to be equal to $\Pi_{ij\in E} w_{ij}$, and we consider the set of players $\mathcal{I}_{Nn}$. We now partition $\mathcal{I}_{Nn}$ into sets of n players. Given the size of the object in hand, it would be appropriate to identify the partitions with the following $|E|$-dimensional space:

$$\Omega = \Pi_{e\in E}[w_e],$$

where $[w_e] = \{1, \cdots, w_e\}$. Specifically, we define a map P from Ω into $2^{\mathcal{I}_{Nn}}$ such that $|P(1,\cdots,1)| = \mathcal{I}_n$, $|P(\omega)| = n$ for all ω, and $P(\omega) \cap P(\omega') = \emptyset$ for $\omega \neq \omega'$. Then the collection $\{P(\omega) : \omega \in \Omega\}$ is a partition of $\mathcal{I}_{Nn}$. For each ω, we define a bijection ϕ_ω from $\mathcal{I}_n$ into $P(\omega)$, such that $\phi_{(1,\cdots,1)}$ is the identity map. We now define a graph G over the vertex set $\mathcal{I}_{Nn}$ with an empty edge set E. Then for each ij that belongs to $E(W)$, for all ω in $\Pi_{e\in E\setminus ij}[w_e]$, m and m' in $[w_{ij}]$ (non-necessarily distinct), we let $\phi_{(\omega,m)}(i)\phi_{(\omega,m')}(j)$ belongs to E. We finally construct a threshold distribution k' on $\mathcal{I}_n$ in such a way that k' equal $k' \circ \phi_\omega^{-1}$ on $P(\omega)$ for all ω in Ω.

To put a note on the construction, given any node i in $\mathcal{I}_n$, suppose j is a neighbor of i, with weight w_{ij} on the edge. For any ω in Ω, $phi_\omega(i)$ has threshold k_i is connected to exactly w_{ij} nodes having thresholds k_j. This large graph is interconnected such that if we restrict the space of action configurations accordingly, the update are locally equivalent.

To this end, let us define the extension map α from $\mathcal{A}_n$ into $\mathcal{A}_{Nn}$ in such a way that for all a in $\mathcal{A}_n$, for all ω and i in $P(\omega)$,

$$(\alpha a)_i = a_{\phi_\omega^{-1}(i)}.$$

It can be checked that for all a in $\mathcal{A}_n$:

$$\alpha(G_k a) = G'_{k'}(\alpha a)$$

The map α is injective, and following the same reasoning as in the proof of Lemma 4.4.9, the result then follows since any cycle in $CYCLE_{Nn}(G',k')$ has cardinality at most equal to 2 by Theorem 4.4.16. □

We move on to extend the result to multigraphs.

4.6 On Convergence Cycles for Multigraphs

We shall not provide a formal description of the model, but rather build on the description provided in the previous section. Given a graph G in $\mathcal{G}_n$ we allow $E(G)$ to contain self-loops, and for every edge ij in E we assign to it a positive weight $w_{ij} = w_{ji}$. With a similar reasoning as in Lemma 4.5.1, without any loss of generality we assume that the weights are integers. We do not provide a formal proof for this fact. To sketch the idea, make a duplicate G' of G, and connect each node i in G with its copy in G' putting on the edge a weight w_{ii}. Remove all self-loops in the new graph to get the model defined in the previous section and apply Lemma 4.5.1.

This done, we then define $\hat{\mathcal{G}}_n$ to be the set of all connected multigraphs over the vertex set $\mathcal{I}_n$. Given a multigraph G in $\hat{\mathcal{G}}_n$, and a node i in $\mathcal{I}_n$, we redefine the neighborhood $\mathcal{N}_i$ of i be a multiset where each node in $\mathcal{N}_i$ has multiplicity equal to the number of edges connecting it to i. In this case, the rule of Proposition 3.1.1 governs the dynamics in this setting.

Theorem 4.6.1. *For any positive integer n, any multigraph G in $\hat{\mathcal{G}}_n$ and any threshold distribution k in $\mathcal{K}_n$, every cycle C in $CYCLE_n(G,k)$ has a cardinality less than or equal to* 2.

Proof. Given a pair (G,k) in $\hat{\mathcal{G}}_n \times \mathcal{K}_n$ such that G contains at least one self-loop, let (G',k') be its bipartite expansion. Then (G',k') contains no self-loops and is necessarily connected (because of the self-loop). We can easily extend from previous lemmas that if every cycle C in $CYCLE_{2n}(G',k')$ has a cardinality less than or equal to 2, then every cycle C in $CYCLE_n(G,k)$ has a cardinality less than or equal to 2. But G' contains only multiple edges, which is equivalent to having positive weights on single edges. The result then follows by Theorem 4.5.2. □

We proceed to discuss convergence time in the following chapter.

Chapter 5

On Convergence Time

This chapter studies the following problem: given a graph G in $\mathcal{G}_n$, a threshold distribution k in $\mathcal{K}_n$ and an initial action configuration a, how many times do we need to iteratively apply G_k on a to reach some cycle C in $CYCLE_n(G,k)$? Recall that for every positive integer n, and every (G,k,a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$, we define $\delta_n(G,k,a)$ to be equal to the smallest non-negative integer T such that there exists a cycle C in $CYCLE_n(G,k)$ and b in C with $G_k^T a = b$. The quantity $\delta_n(G,k,a)$ denotes to the minimal number of iterations needed until a given action configuration a reaches a cycle, when iteratively applying G_k. We refer to $\delta_n(G,k,a)$ as the *convergence time* from a under G_k. We begin by showing that there exists some positive integer m, such that for every positive integer n, and every (G,k,a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$, the convergence time $\delta_n(G,k,a)$ is less than or equal to mn^2. We then proceed to improve the bound to being equal to be linear in the size of the network for some graph structure cases. Formally, for all positive integers n, and every (G,k,a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ where G is a cycle graph, a complete graph or a tree, the convergence time $\delta_n(G,k,a)$ is less than or equal to n. We end the chapter by discussing tighter bound on the convergence time over general graphs.

5.1 On Quadratic Time over General Graphs

As a quick follow-up of the infrastructure built in the section on convergence cycle, we have the following bound:

Theorem 5.1.1. *For all positive integers n, and every (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$, the convergence time $\delta_n(G, k, a)$ is less than or equal to mn^2 for some positive integer m.*

Proof. Given a positive integer n, let (G, k) be a point in $\mathcal{G}_n \times \mathcal{K}_n$, let (G', k') be the symmetric-expansion of (G, k) in $\mathcal{S}_{n'}$, and let (G'', k'') be the bipartite-expansion of (G', k') in $\mathcal{S}_{2n'}$. We have the following fact:

$$\delta_n(G, k) \leq \delta_{n'}(G', k') \leq \delta_{2n'}(G'', k'').$$

Moreover, we have that:

$$\delta_{2n'}(G'', k'') \leq \max_{a \in \mathcal{A}_{2n'}} |E_c^{G''}(a)|,$$

The fact follows from the fact that $G_k|_{P_o} b \neq b$ if and only if $E_c(G_k|_{P_o} b) < E_c(b)$ and $G_k|_{P_e} b \neq b$ if and only if $E_c(G_k|_{P_e} b) < E_c(b)$. Additionally for all a in $\mathcal{A}_{2n'}$ we have:

$$|E_c^{G''}(a)| \leq |E''| \leq 2|E'| \leq 2[n^2 + 3 \sum_{i \in \mathcal{I}_n} b_i + w_i],$$

where E' and E'' denotes the set of edges of G' and G'' respectively. Finally:

$$\sum_{i \in \mathcal{I}_n} b_i + w_i = \sum_{i \in \mathcal{I}_n} d_i + 1 = 2|E| + n.$$

The result follows. □

The constant m in the theorem statement can be optimized, but it is of no interest. Instead it would be interesting to prove a bound below quadratic. One thing to notice from the proof above is that if the graph has bounded degrees, the convergence time is less than a linear function of the size of the network. We turn back to the cases of cycle graphs, complete graphs and trees and derive tighter upper bounds.

5.2 On Linear Time over Cycle Graphs

We shall restrict the analysis in the section to *even* positive integers n. In this case, every 2-regular connected graph in $\mathcal{G}_n$ is bipartite and we make use of the bipartite property. Let G be cycle graph in $\mathcal{G}_n$. Recall from Section 4.1 that we defined s and p to be maps from $\mathcal{I}_n$ into $\mathcal{I}_n$ (we refer to them successor and predecessor) such that for node i in $\mathcal{I}_n$, i and s_i are neighbors, i and p_i are neighbors and $(sp)_i = (ps)_i = i$. In this setting, $(ss)_i$ refers to the successor of the successor of node i and is denoted as s_i^2. Recursively, the notation s_i^m, where m is some non-negative integer, denotes the node obtained by iteratively applying (m times) the successor function s on i. A similar notation holds for the predecessor function p. We now pick a player i in $\mathcal{I}_n$, and consider the subset $\{s_i^{2m} : m \geq 0\}$ of $\mathcal{I}_n$. First, $\{s_i^{2m} : m \geq 0\}$ is not equal to $\mathcal{I}_n$, this follows from the fact that n is even. Furthermore, $\{s_i^{2m} : m \geq 0\}$ is equal to $\{p_i^{2m} : m \geq 0\}$. The implies that the update rule over two time steps of player i depend only on the information available in the actions taken by the players in $\{s_i^{2m} : m \geq 0\}$. We consider a 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to G, and point out that the sequence $a, G_k^2 a, G_k^4 a, \cdots$ is constant after time step $2T$ if and only if both sequences $a\restriction P_o, G_k^2 a\restriction P_o, G_k^4 a\restriction P_o, \cdots$ and $a\restriction P_e, G_k^2 a\restriction P_e, G_k^4 a\restriction P_e, \cdots$ are constant after time step $2T$.

Proposition 5.2.1. *For all positive even integers n, and every (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ where G is 2-regular, the convergence time $\delta_n(G, k, a)$ is less than or equal to n.*

Proof. Let us then consider the sequence $a\restriction P_o, G_k^2 a\restriction P_o, G_k^4 a\restriction P_o, \cdots$, we define T to be the minimum integer m such that $G_k^{2m} a = G_k^{2m+2} a$. We claim that the integer T is less than or equal to the cardinality of P_o. Let $S(b)$ be the number of players playing the strong assignment in action configuration b, then if $b \neq G_k^2 b$ then $S(b) < S(G_k^2 b)$ for all action configuration b. But $G_k^{2m} a \neq G_k^{2m+2} a$ for all $m < T$ and so $P_o \geq S(G_k^{2T} a) \geq S(a) + T$. But $P_o = n/2$, and so the result follows. $\square$

In other words, as long as the sequence is not constant, one player is switching his action over two time steps. However, every player has a strong assignment (non-valid nodes trivially have a strong assignment), and so each player is allowed to flip only once over two time steps if ever. After $2T$ time steps, no player is able to flip over two time steps.

5.3 On Linear Time over Complete Graphs

Recall that for a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is complete, $\mathcal{B}_k$ is the subset of $\mathcal{A}_n$ such that a belongs to $\mathcal{B}_k$ if and only if for every player i in $\mathcal{I}_n$, if player i plays $\mathbb{B}$ in a, then each player j with $k_j < k_i$ plays $\mathbb{B}$ in a.

Definition 5.3.1. *It is said that $a \in \mathcal{B}_k$ is at level L (or L is the level of a) if and only if at least some node i in $\mathcal{I}_n$ having $k_i = L$ is playing $\mathbb{B}$ in a and every node having $k_j > L$ is playing $\mathbb{W}$ in a. If a is $\mathbb{W}$ everywhere on $\mathcal{I}_n$, we say that a is at level -1.*

Proposition 5.3.2. *For any pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is complete, every action configuration in $G_k(\mathcal{A}_n)$ is at level L for some L in $\{-1\} \cup \mathbb{N}$.*

Proof. The result follows from Proposition 4.2.1, i.e. $G_k(\mathcal{A}_n) \subset \mathcal{B}_k$. □

Proposition 5.3.3. *For any pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is complete and any configuration a in $\mathcal{A}_n$, if $G_k a$ is at level -1, then $G_k^m a$ is $\mathbb{W}$ for all positive integers m.*

Proof. If $G_k a$ is at level -1, then $G_k a$ is equal to $\mathbb{W}$ everywhere on $\mathcal{I}_n$ and all nodes have a positive threshold. Therefore, $G_k^2 a$ is also at level -1. The result follows by induction. □

Proposition 5.3.4. *For any pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is complete and any configuration a in $G_k(\mathcal{A}_n)$, if a, $G_k a$ and $G_k^2 a$ are all at level L, then $G_k^m a$ is at level L for all non-negative integers m.*

Proof. We will suppose that neither a not $G_k a$ is not a fixed point of G_k, otherwise the statement trivially holds. It follows that a, $G_k a$ and $G_k^2 a$ are all at level L only if

$|k^{-1}(L)|$ is even, $L = |k^{-1}(L)|/2$ and $|k^{-1}(L) \cap a^{-1}(\mathbb{B})| = |k^{-1}(L) \cap a^{-1}(\mathbb{W})|$. In that case, we get $|k^{-1}(L) \cap G_k a^{-1}(\mathbb{B})| = |k^{-1}(L) \cap G_k a^{-1}(\mathbb{W})|$ and by induction we obtain $|k^{-1}(L) \cap (G_k^m a)^{-1}(\mathbb{B})| = |k^{-1}(L) \cap (G_k^m a)^{-1}(\mathbb{W})|$ for all non-negative integers m. □

Proposition 5.3.5. *For any pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is complete and any configuration a in $G_k(\mathcal{A}_n)$, consider the sequence $L_0, L_1, L_2, \cdots$ where L_m is the level of $G_k^m a$. Then $L_{n-1} = L_{n-1+m}$ for all non-negative integers m.*

Proof. Without any loss of generality we may assume that a is at level $L \neq -1$. If condition bla happens then we are done. So suppose such a condition is not there, then either $L_0 \neq L_1$ or $L_0 = L_1 \neq L_2$. If $L_0 \neq L_1$, then by monotonicity of action configurations, the sequence $L_0, L_1, L_2, \cdots$ starts as strictly monotone, then becomes constant at some L. However the sequence $L_0, L_1, L_2, \cdots$ can contain at most n levels since we have n nodes, and so $L_{n-1} = L_{n-1+m}$ for all non-negative integers m. If $L_0 = L_1 \neq L_2$, then by monotonicity of action configurations, the sequence $L_1, L_2, \cdots$ starts as strictly monotone, then becomes constant at some L. However, $k^{-1}(L_0)$ contains more than one node, otherwise either $L_0 \neq L_1$ or the sequence is constant. Therefore, there are at most $n-1$ levels and so the sequence $L_1, L_2, \cdots$ can contain at most $n-1$ levels, therefore $L_{n-1} = L_{n-1+m}$ for all non-negative integers m. □

Proposition 5.3.6. *For all positive even integers n, and every (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ where G is complete, the convergence time $\delta_n(G, k, a)$ is less than or equal to n.*

Proof. If a belongs to $G_k(\mathcal{A}_n)$, then $\delta_n(G, k, a) \leq n-1$ by the previous proposition. To get the result, we need to apply G_k once on an action configuration to get an action configuration in $G_k(\mathcal{A}_n)$. □

If the sequence $L_0, L_1, L_2, \cdots$ or $L_1, L_2, \cdots$ starts as strictly monotone then at n it has reached a fixed point by monotonicity. If the sequence is constant, then it is either at a cycle of length 2 or at a fixed point.

5.4 On Linear Time over Trees

In this section, the letter T shall always be used to denote trees, and never time as was done sometimes in previous sections. We shall prove a lemma, and then derive our result from it. We begin by a definition.

Definition 5.4.1. *Given a triplet* (G,k,a) *in* $\mathcal{G}_n^b \times \mathcal{K}_n \times \mathcal{A}_n$ *and a 2-Partition* (P_o, P_e) *of* $\mathcal{I}_n$ *with respect to* G. *It is said that* $a\restriction P_o$ *is reachable in* (G,k) *if there exists* a' *in* $\mathcal{A}_n$ *such that* $a\restriction P_o = (G_k|_{P_o}a')\restriction P_o$. *In this case, it is said that* $a'\restriction P_e$ *induces* $a\restriction P_o$. *Similarly,* $a\restriction P_e$ *is reachable in* (G,k) *if there exists* a' *in* $\mathcal{A}_n$ *such that* $a\restriction P_e = (G_k|_{P_e}a')\restriction P_e$. *And again, it is said that* $a'\restriction P_o$ *induces* $a\restriction P_e$.

Proposition 5.4.2. *Given a triplet* (G,k,a) *in* $\mathcal{G}_n^b \times \mathcal{K}_n \times \mathcal{A}_n$ *and a 2-Partition* (P_o, P_e) *of* $\mathcal{I}_n$ *with respect to* T. *If node* i *in* P_e *is non-valid and* $a\restriction P_e$ *is reachable in* (G,k), *then* $((G_k|_{P_e}G_k|_{P_o})^m a)_i = a_i$ *for all non-negative integers* m.

Proof. The proposition is rather trivial and follows from the definition of non-validity and reachability. □

Proposition 5.4.3. *Given a triplet* (T,k) *in* $\mathcal{G}_n^b \times \mathcal{K}_n$ *where* T *is a tree, and a 2-Partition* (P_o, P_e) *of* $\mathcal{I}_n$ *with respect to* T. *Pick a node* r *to be the root of* T, *then if player* i *in* P_e *has only leaves as children (with respect to* r*) and* $1 < k_i < d_i$ *then* $((T_k|_{P_e}T_k|_{P_o})^m a)_i = a_i$ *for all non-negative integers* m.

Proof. In the case where $1 < k_i < d_i$, both actions, $\mathbb{W}$ and $\mathbb{B}$, are strong assignments for node i. □

Proposition 5.4.4. *Given a triplet* (T,k,a) *in* $\mathcal{G}_n^b \times \mathcal{K}_n \times \mathcal{A}_n$ *where* T *is a tree, and a 2-partition* (P_o, P_e) *of* $\mathcal{I}_n$ *with respect to* T. *Suppose* $a\restriction P_o$ *is reachable and* $a\restriction P_o$ *induces* $a\restriction P_e$ *both in* (T,k). *Then if* P_e *consists of only one node, that node will never change its action.*

Proof. We suppose that the node in P_e is valid, otherwise nothing is to be done. Furthermore, without any loss of generality we may assume the nodes in P_o to also be valid, otherwise they will never switch actions given that $a\restriction P_o$ is reachable, and so we may remove them from the network. It follows that both $\mathbb{B}$ and $\mathbb{W}$ are strong assignments for the node in P_e. $\square$

Lemma 5.4.5. *For every positive integer n, given a triplet (T, k, a) in $\mathcal{G}_n^b \times \mathcal{K}_n \times \mathcal{A}_n$ where T is a tree and a 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to T. If $a \restriction P_o$ is reachable and $a \restriction P_o$ induces $a \restriction P_e$ both in (T, k), then there exists a player i in P_e, such that $((T_k|_{P_e} T_k|_{P_o})^m a)_i = a_i$ for all non-negative integers m.*

Proof. Given the nice structure the tree possesses, there are several ways we can perform the induction. We shall proceed by induction on the number of nodes in the tree. We start with the base case that refers to a two node graph with a single edge. In this setting, there are only six cases of possible threshold distribution. It is fairly straightforward to exhaustively check them, so we omit the proof for $n = 2$. We suppose that the statement holds for graphs with n nodes, and we show that it holds for graphs with $n + 1$ nodes.

We pick a triplet (T, k, a) in $\mathcal{G}_n^b \times \mathcal{K}_n \times \mathcal{A}_n$ where $T(\mathcal{I}_n, E)$ is a tree, and a 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to T. We suppose that $a \restriction P_o$ is reachable and $a \restriction P_o$ induces $a \restriction P_e$ both in (T, k). If there exists a player in P_e that is non-valid with respect to (G, k), the statement trivially holds by Proposition 5.4.2. We will assume that all nodes P_e are valid nodes. We may also assume that all nodes in P_o are valid nodes, otherwise they would never change actions and so can be removed. We pick a node r in P_o to be the root of the tree. If there exists some player i in P_e that has only leaves as children (with respect to r) and $1 < k_i < d_i$, then the statement holds by proposition 5.4.3. Then we will assume that no such player exists. Moreover, if a player in P_e is playing a strong assignment, then the statement trivially holds. So, we shall assume that no player is playing a strong assignment. And finally, if P_e contains only one node, then the statement holds by Proposition 5.4.4. We shall then assume that P_e contains at least two players.

We argue by contradiction. Suppose that for every player i in P_e, there exits a positive integer M_i such that $((T_k|_{P_e} T_k|_{P_o})^{M_i} a)_i \neq a_i$ and $((T_k|_{P_e} T_k|_{P_o})^m a)_i = a_i$ for $m < M_i$. We set M_f to be $\max_i M_i$, and pick a node e in P_e such that $M_e = M_f$. We consider an edge eo in E such that $a_o = a_e$ and if e has only leaves as children, the connected components of $(\mathcal{I}_n, E\backslash\{oe\})$ contain at least one node in P_e. Such an edge always exists given what we assumed earlier.

We then construct a pair (T', k') as follows. Define T' to be the connected component of the graph with vertex set $\mathcal{I}_n$ and edge set $E\backslash\{oe\}$ not containing e. Set k' to be equal to k on $V(T')\backslash\{o\}$, k on $\{o\}$ if $a_e = \mathbb{W}$, and $(k-1)\vee 0$ on $\{o\}$ if $a_e = \mathbb{B}$.

One can check that $a\restriction(P_o \cap V(T'))$ is reachable and $a\restriction(P_o \cap V(T'))$ induces $a\restriction(P_e \cap V(T'))$ both in (T', k'). In this case, there exists a player i in $P_e \cap V(T')$, such that $((T'_{k'}|_{P_e} T'_{k'}|_{P_o})^m a)_i = a_i$ for all non-negative integers m. Then $((T_k|_{P_e} T_k|_{P_o})^m a)_i = a_i$ for all positive integer m such that

$$((T_k|_{P_e} T_k|_{P_o})^{m-1} a)_e = a_e,$$

that is for $m \le M_f$. Then, player i *can* only flip in (T, k) after M_f, contradicting the definition of M_f. □

Proposition 5.4.6. *For all positive integers n, and every (T, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ where T is a tree, the convergence time $\delta_n(T, k, a)$ is less than or equal to n.*

Proof. Let T be a tree in $\mathcal{G}_n$, and consider a 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to T such that $|P_e| \le |P_o|$. For any k in $\mathcal{K}_n$ and a in $\mathcal{A}_n$, if we consider $(a^o, a^e) = (T_k a\restriction P_o, T_k^2 a\restriction P_e)$, then a^o is reachable, and a_e induces a_o. Then by Lemma 5.4.5, there exists at least one node in P_e such that $((T_k|_{P_e} T_k|_{P_o})^m (a^o, a^e))_i = a_i^e$ for all non-negative integers m. Let T' be a connected component of the induced subgraph of T over $\mathcal{I}_n\backslash\{i\}$ such that $|V(T') \cap P_e|$ is maximal. Define k' to be equal to $k\restriction V(T')$ on $V(T')\backslash\mathcal{N}_i$, equal to k on $V(T') \cap \mathcal{N}_i$ if $a_i^e = \mathbb{W}$, and equal to $(k-1)\vee 0$ on $V(T') \cap \mathcal{N}_i$ if $a_i^e = \mathbb{B}$. Then,

$$\delta_n(T, k, a) = 2 + \delta(T, k, T_k^2 a) = 2 + \delta(T', k', (T_k^2 a)\restriction V(T')) \le 2|P_e| \tag{5.1}$$

by successive application of Lemma 5.4.5. But since $|P_e| \le |P_o|$, we get $|P_e| \le n/2$ and the result follows. □

Notice that leaving the bound at $2|P_e|$ in the preceding proof gives a better bound that is equal to twice the size of a minimal vertex cover. We proceed to discuss extending such results to the general case.

5.5 On Linear Time over General Graphs

Unfortunately, Lemma 5.4.5 does not hold over all bipartite graphs. We shall provide a counterexample, namely we give a triplet (G, k, a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ and a 2-Partition (P_o, P_e) of $\mathcal{I}_n$ with respect to G, such that $a \restriction P_o$ is reachable and $a \restriction P_o$ induces $a \restriction P_e$ both in (G, k), but for each player i in P_e, there exists a non-negative integer m such that $((G_k|_{P_e} G_k|_{P_o})^m a)_i \neq a_i$.

To this end, consider we consider the set $\mathcal{I}_{2^n-1}$ for some *odd* n large enough. Let $T(\mathcal{I}_{2^n-1}, E)$ be a complete binary tree over $\mathcal{I}_n$. Let L be the subset of T containing the leaves, and let r be the root. We construct a new edge set $E' = E \cup_{l \in L} \{lr\}$ and consider the graph $G(\mathcal{I}_{2^n}, E)$. Since n is odd, then G is bipartite. We set k to be equal to 1 on $\{r\} \cup L$ and equal to 2 elsewhere. We consider a two partition (P_o, P_e) on $\mathcal{I}_{2^n-1}$ with respect to G such that r belongs to P_o. We say that node i is at depth D if and only if the edge path from r to i consists of D edges. Then, all the leafs are at depth $n-1$.

We now consider the action configuration a' in $\mathcal{A}_{2^n-1}$ that is equal to $\mathbb{B}$ on all nodes at depth $n-3$, and $\mathbb{W}$ everywhere else. Consider $a = G_k|P_e G_k|P_o a'$, then $a \restriction P_o$ is reachable and $a \restriction P_o$ induces $a \restriction P_e$ both in (G, k). The action configuration a is $\mathbb{B}$ on all nodes at depth $n-5$ and $\mathbb{W}$ everywhere else.

We now claim that for each player i in P_e, there exists a non-negative integer m such that $((G_k|_{P_e} G_k|_{P_o})^m a)_i \neq a_i$. To show this, we will claim three things that can be easily checked. First, if all the nodes at depth $n-1$ are $\mathbb{B}$ in $(G_k|_{P_e} G_k|_{P_o})^m a$ for some m, then all nodes will play $\mathbb{B}$ at the limit. Second, for each node i, there exists a non-negative integer m such that $((G_k|_{P_e} G_k|_{P_o})^m a)_i = \mathbb{W}$. We can simply consider $m = 0$ and $m = 1$ where $m = 0$ corresponds to a, and $m = 1$ corresponds to $G_k|P_e G_k|P_o a$ that is equal to $\mathbb{B}$ only on nodes at depth $n-7$. Third, all the nodes at depth $n-1$ are $\mathbb{B}$ in $(G_k|_{P_e}G_k|_{P_o})^m a$ for some m. It is easy to see that there exists an m such that $(G_k|_{P_o}(G_k|_{P_e}G_k|_{P_o})^{m-1}a)_r = \mathbb{B}$, then $G_k|_{P_e}G_k|_{P_o}(G_k|_{P_e}G_k|_{P_o})^{m-1}a$ will be $\mathbb{B}$ on L since $k=1$ on L.

Although this disproves that the lemma holds for all bipartite graphs, the convergence time in this counterexample is still not more than n. We end this section with a rough conjecture that the convergence time for bipartite graph is less than or equal to n. However, in this case the convergence time for general non-bipartite graphs is not more than $2n$. To have that bound, we claim that the convergence time for a pair (G,k) is at most that of its bipartite expansion.

We finally show that if $2n$ is an upper bound, it is rather tight. It is possible to construct an instance of (G,k,a) in $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ that has a convergence time of $2n-3$. Indeed, we consider the set $\mathcal{I}_n$ and construct a graph G as follows. We connect $n-2$ players in a line, pick one of the nodes having degree 1, connect the remaining two players to it, and connected those two players together (one gets a line ending with a triangle). We set k in K_n to be equal to 1 over $\mathcal{I}_n$, and a is $\mathbb{B}$ on the only player with degree 1, and $\mathbb{W}$ everywhere else. One can check that $2n-3$ steps are needed to reach a cycle. In this case, the cycle consists of one action profile.

5.6 On Convergence Time of the Extension Model

In this section, we show that the conjecture on convergence time being at most n steps for bipartite graphs would fail if we allow weighted edges for the graphs. To show that, we consider an example induced by the counterexample in the last section by simply collapsing the node on the same level. We consider a pair (W,k) in $\mathcal{W}_6 \times \mathcal{K}_6$ where W is a weighted graph over $\mathcal{I}_6 = \{i_1, i_2, i_4, i_8, i_{16}, i_l\}$, with edge set $\{i_1i_2, i_2i_4, i_4i_8, i_8i_{16}, i_{16}i_l, i_li_1\}$, thresholds $(k_{i_1}, k_{i_2}, k_{i_4}, k_{i_8}, k_{i_{16}}, k_{i_l}) = (1,2,4,8,16,1)$ and weights $(w_{i_1i_2}, w_{i_2i_4}, w_{i_4i_8}, w_{i_8i_l}, w_{i_{16}i_l}, w_{i_li_1}) = (1,2,4,8,16,1)$.

Finally, consider the action configuration a in $\mathcal{A}_5$ equal to $\mathbb{B}$ on i_4 and $\mathbb{W}$ everywhere else. We get the following sequence of action profiles:

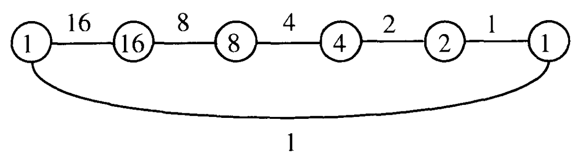

Figure 5-1: Partial visualization of the pair (W, k).

	i_1	i_2	i_3	i_4	i_5	i_l
T.0	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$
T.1	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{W}$
T.2	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$
T.3	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{B}$
T.4	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$
T.5	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{B}$
T.6	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{W}$	$\mathbb{W}$
T.7	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{B}$
T.8	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$	$\mathbb{B}$	$\mathbb{W}$

Notice, that we arrived to a cycle in 7 steps instead of 6. Furthermore, increasing the size of the network would yield a bigger gap asymptotically reaching $2n$.

It is important to clarify that we did not actually establish quadratic convergence time for the extension model allowing weighted edges. In fact, the construction provided in Chapter 4 to prove the length of convergence cycles for weighted-edge graphs cannot be used to extend the quadratic convergence time. But we shall not explicitly tackle this question, instead we keep the problem to a rough conjecture. Although the counterexample provided above disproves the convergence time in at most n step, linear convergence time is still plausible for the extension model allowing weighted edges. In that case, a linear convergence time for multigraphs would also follow.

We now proceed to characterize the number of limiting cycles.

Chapter 6

On the Complexity of Counting

So far, we have been dealing with bounds that are uniform over all graphs, all thresholds and all action configuration. The natural coming step would be to find bounds on the number of cycles (fixed-points and non-degenerate cycles), the number of fixed points and the number of non-degenerate cycles for all graphs G in $\mathcal{G}_n$ and all threshold distributions k in $\mathcal{K}_n$. We let $\overline{F}$ and $\underline{F}$ be respectively the maximum and minimum number of fixed-points over all pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$. Likewise, let $\overline{C}$ and $\underline{C}$ be respectively the maximum and minimum number of non-degenerate cycles over all pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$.

Proposition 6.0.1. *The lower bound $\underline{F}$ is upper bounded by* 2.

Proposition 6.0.2. *The lower bound $\underline{C}$ is equal to* 0.

Proof. To prove those two proposition, it would be enough to consider any 2-connected regular graph of n players where n is odd, and provide each player with a threshold equal to 1. All players playing $\mathbb{B}$ and all players playing $\mathbb{W}$ are the only fixed-points. No non-degenerate cycles exist at the limit. □

Proposition 6.0.3. *The upper bound $\overline{F}$ is lower bounded by $2^{n/3}$.*

Proposition 6.0.4. *The upper bound $\overline{C}$ is lower bounded by $2^{n/3} - 1$*

Proof. Consider the 2-connected regular graph of n players where n is a multiple of 3 not equal to 3. Assign $n/3$ nodes a threshold of 2 and $2n/3$ nodes a threshold of 1 in such a way that each node of threshold 1 has exactly one neighbor of threshold 1 connected to it. In this case, we have $n/3$ pairs of neighbors having thresholds of 1, and so we can construct at least $2^{n/3}$ fixed points where the neighbors in each pair are either playing both $\mathbb{B}$ or both $\mathbb{W}$. We can similarly construct at least $2^{n/3} - 1$ non-degenerate cycles by having for each pair of such neighbors, either both neighbors as $\mathbb{W}$ or exactly one of the neighbors as $\mathbb{B}$. $\square$

We are not concerned about exact bounds, those claim serve only to show that we are dealing with a rather wide range of number of limiting outcomes. This said, we will study whether we can have an arbitrarily good characterization of the count. Instead of providing bounds, we will study how tractable is it to count equivalence classes, fixed points and non-degenerate cycles. Ultimately, we show that those counting problems are #P-Complete. We begin by providing a quick review of complexity theory.

6.1 On Languages and Turing Machines

We shall not delve deeply into defining and presenting the basic concept, we refer the reader to [17] for a thorough exposition.

Given an alphabet $\Sigma = \{0, 1\}$, we refer to Σ^* as the set of all binary strings constructed from elements of Σ.[1] The set Σ^* may be identified with $\cup_{n=0}^{\infty}\Sigma^n$ where Σ^0 contains the empty set.

Definition 6.1.1. *A language is a subset of* Σ^*.

Given a language A, we aim to design a unit (machine) such that given a string s in Σ^*, this unit is able to decide whether s belongs A or not. Let us consider the case of a generic yes/no question. Suppose a language A consists of the instances of the question where the answer is yes, and suppose some unit M accept strings

[1]The unary operator $*$ is called the Kleene star.

representing the questions that admit yes as an answer i.e. accepts only strings that are in A, then this unit can basically answer our questions.

To this end, we use the model of a Turing machine (TM) as a computation device, and we settle on an informal description of it; this is more than enough for our purpose. The Turing machine will only serve as a precise model for the definition of algorithm as intuitively understood.

A **(deterministic) Turing machine** consists of a finite state machine, an infinite tape partitioned into an infinite number of consecutive cells (a cell can have at most two adjacent cells one on its right and one on its left) and a read/write head positioned on that tape over a cell.

We first consider the tape: each cell in the tape may contain a symbol in Σ.[2] Furthermore, we assume (without any loss of generality) that the tape is *one-sided* in the sense that every cell in the tape has finitely many cells on its left and infinitely many cells on its right. Second, the finite state machine contains a starting state, an accept state and a reject state. Third, the TM is allowed to both write on the tape or read from it (one cell at a time), and the read/write head is allowed to move left or right along the tape.

We initialize the Turing Machine as follows. We require the TM to be fed a string in Σ^* as input, and so we initially scribe that string on the tape where the first symbol is written in the leftmost cell, and the other symbols are written in order (each in one cell) as we move right along the tape. We initialize the finite state machine to be in the initial state, and we set the read/write head to be positioned on the leftmost cell. We will simply refer to the TM as being in a state instead of explicitly mentioning the finite state machine.

We now run the TM. The TM reads the symbol in the cell where the head is positioned. Depending on the current state and the symbol read, it writes a new symbol (replacing the old symbol) on the cell where the head is positioned, transitions to a new state and moves the head either left or right by one cell. Put differently, the

[2]For convenience, it would be necessary to define a 'blank' symbol to be written in a cell when that cell is supposed to contain no symbol in Σ. However, we shall not worry about this issue since we keep the description rather informal.

TM is governed by a transition function takes a pair of state and symbol as input, and outputs a triplet consisting of a state, a symbol and a move direction for the head. The TM repeats this procedure moving from state to state. As mentioned, there are two special states: the accept state and the reject state. If the machine reaches any of them, it halts. If the machine reaches the accept state, it accepts the input. If the machine reaches the reject state, it rejects the input.

However, we note that a Turing Machine need not necessarily reach an accept state or a reject state. In this case, it never halts.

Definition 6.1.2. *A Turing Machine M is said to decide a language A if and only if M accepts every string in A and rejects every string not in A. A language is said to be decidable if there exists a TM that decides it.*

We shall restrict the description to languages that are only decidable, in the sense that there exists a Turing machine that halts on every input, and accepts a string if and only if it is in the language. Our need for the Turing Machine stands as to have it solve problems and answer questions for us. However, we note that Turing Machines always takes a string as input, so we are required to encode our questions and problems as string. A decoder can then be built-in within the Turing Machine as part of its states, transition rules, and read/write operations. We proceed to define the central notion of time complexity.

Definition 6.1.3. *Let M be a deterministic Turing machine that halts on all input strings. The running time or time complexity of M is a function f from $\mathbb{N}$ into $\mathbb{N}$ such that $f(n)$ is the maximum number of steps that M uses on any input of length n.*

If $f(n)$ is the running time of M, we say that M runs in time $f(n)$ and that M is an $f(n)$ time Turing Machine.

Finally, a variant of the deterministic TM model is the **non-deterministic Turing machine** (NTM). Building on the previous description, for every symbol read on the tape and every state the finite state the NTM is in, the NTM can branch out to multiple possibilities and check them at the same time. Going back to the transition function perspective as considered earlier, the NTM is governed by a transition function takes a pair of state and symbol as input, and outputs a collection of triplets each consisting of a state, a symbol and a move direction for the head. The transition function is then applied to the pair of state and symbol in each of those triplets. This said, the NTM has simultaneous computation paths. If one of the path leads the NTM to an accept state, the machine accepts the input.

We go on defining both decision and function problems in the following sections.

6.2 On Decision Problems

Decision problems are questions that require yes/no answers. To illustrate, let us consider the following problem: *given a graph $G \in \mathcal{G}_n$ as input, is the graph G bipartite?* This a yes or no question, and the language *is-Bipartite* 'describing' it can be the set of all graphs in $\mathcal{G}_n$ that are bipartite. Notationally,

$$\textit{is-Bipartite} = \{\langle G\rangle : G \in G_n^b\}. \tag{6.1}$$

Given an encoding for graphs (those may be easily encoded by their adjacency matrix) into strings in Σ^*, we may construct a Turing machine that decides *is-Bipartite.*

Definition 6.2.1 (Polynomial Time). *The class P is the class of languages that are decidable in polynomial time on a deterministic Turing machine.*

Equivalently, we can devise a (deterministic) polynomial time algorithm to decide whether an input belongs to the language, i.e. to answer the question. Going back to the bipartite testing case, it is fairly straightforward to polynomially decide whether a graph is bipartite or not. However, also note that G_n^b consists only of connected graphs. Therefore, we also need to test connectedness along the way, but that also can be solved in polynomial time. Therefore, *is-Bipartite* belongs to the class P. It is crucial to note that the algorithms considered in this case are deterministic. We turn to non-determinism, but let us define verifiers first.

Definition 6.2.2. *A verifier for a language A is an algorithm V, such that w is in A if and only if there exists a string c (called certificate) such that V accepts the input $\langle w, c\rangle$. A polynomial time verifier runs in polynomial time in the length of w.*

Building on that definition, we define:

Definition 6.2.3 (Non-Deterministic Polynomial Time). *The class NP is the class of languages that have polynomial time verifiers.*

Going back to the *is-Bipartite* language. Given a graph G, the certificate in that case would be a 2-Partition of the nodes in G and a spanning tree over the nodes in G to prove connectedness. This said, we may devise a polynomial time algorithm such that given a pair $\langle G, (p, T)\rangle$ as input, where G is a graph, p is a partition of the vertex set of G and T is a subgraph of G, accepts the input if only if p is a 2-Partition of G and T is a spanning tree of G. We have that *is-Bipartite* belongs to NP. Equivalently, we may define the class NP as follows.

Proposition 6.2.4. *A language is in NP if and only if it is decided by some nondeterministic polynomial time Turing machine.*

Proof. We shall only give out a sketch of the proof. The idea is rather simple. If a language A is in NP, then it has a polynomial time verifier V. We can construct a TM that non-deterministically goes over all possible certificate, and tests each candidate with V along one branch. Conversely, if a language A is decided by some nondeterministic polynomial time Turing machine, we may construct a polynomial time verifier as follows. Given an input $\langle w, c\rangle$ where both w and c are strings, run the non-deterministic Turing machine treating c as a description of the non-deterministic choices to make. If along those choices the machine accepts w, the algorithm accepts $\langle w, c\rangle$. □

From this, we see that P is a subset of NP. The famed open question is whether NP is a subset of P. Let us give out a different language in NP that is of different nature than *is-Bipartite*. Consider the language CLIQUE to be $\{ \langle G, m \rangle : G$ is an undirected graph having a clique of size m as subgraph $\}$. In this context, given a graph G, a certificate would be a set of m vertices in $V(G)$ forming a clique in G. A polynomial time verifier may be easily devised for this language, and so CLIQUE belongs to NP. The reason why we mention CLIQUE is that some problems in NP have a complexity that is related to the entire class, by this we mean that if a polynomial time algorithm exists for any of these problems, one exists for all the problems in NP. Such problems are called NP-Complete. We formalize the idea.

Definition 6.2.5. *A function f from Σ^* into Σ^* is a polynomial time computable function if some polynomial time Turing machine M exists that halts with just $f(w)$ on its tape, when started on any input w.*

Definition 6.2.6. *Language A is polynomial time mapping reducible, or simply polynomial time reducible, to language B, written $A \leq_P B$, if a polynomial time computable function f from Σ^* into Σ^* exists such that $w \in A$ if and only if $f(w) \in B$. The function f is called a polynomial time reduction of A to B.*

Definition 6.2.7. *A language B is NP-hard if every A in NP is polynomial time reducible to B.*

Definition 6.2.8. *A language B is NP-complete if B is in NP and B is NP-Hard.*

Usually, to check that a language C is NP-complete, we do not directly show that every problem in NP may be reduced to C. We make use of the following:

Theorem 6.2.9. *If B is NP-complete and $B \leq_P C$ for C in NP, then C is NP-complete.*

Proof. The only fact that needs to be checked is that every A in NP is polynomial time reducible to C. Let A be a language in NP. Since B is NP-complete, then A is polynomial time reducible to B. Let f be a polynomial time reduction of A to B. We also know that $B \leq_P C$. Let g be a polynomial time reduction of B to C, the composition $g \circ f$ is then a polynomial time reduction of A to C. $\square$

However to be able to use that theorem, we need a repertoire of NP-hard problems. We present some:

Definition 6.2.10. *The language SAT is the collection of boolean formulas that have a satisfying assignment.*

The following is due to the work of S. Cook and L. Levine in the early 1970s:

Proposition 6.2.11. *SAT is NP-Complete.*

Deciding satisfiability remains hard even if we restrict the set of boolean formulas as follows:

Definition 6.2.12. *A boolean formula is in Conjunctive Normal Form (CNF) if it is a conjunction of disjunction clauses. Formally, let $x_1, \cdots x_n$ be boolean variables, a boolean formula ϕ is in CNF-form if ϕ is written as*

$$\phi(x_1, \cdots, x_n) = \bigwedge_{i=1}^{m} (y_{i,1} \vee \cdots \vee y_{i,k_m})$$

where for each m, k_m is a positive integer, and $y_{.}$ represents either x_i of $\neg x_i$ for some i.[3] *For each possible i and j, $y_{i,j}$ is called a* literal. *For each m, the disjunction of literals $(y_{i,1} \vee \cdots \vee y_{i,k_m})$ is called a clause.*

We may define the following problem:

Definition 6.2.13. *The language $3SAT$ is the collection of boolean formulas having a satisfying assignment, that which in CNF consist of clauses having at most three literals.*

Proposition 6.2.14. *3SAT is NP-Complete.*

We proceed to overview function problems.

[3]Let x be a boolean variable, then $\neg x$ is the negation of x, i.e. $\neg x = 1$ if and only if $x = 0$.

6.3 On Function and Counting Problems

Function problems would roughly include problems of the form: given some input w, compute $f(w)$. This said, it would be useful to keep in mind a different characterization of functions in general. We define a rule of assignment is to be a relation r subset of some product set $C \times D$ where for every c in C there exists a unique d in D such that (c, d) belongs to r. The set C is considered to be the domain of r, the set $\{d \in D : (c, d) \in r \text{ for some } c \text{ in } C\}$ is said to be the range of r. A function f is then a rule of assignment r along with a set B containing the image set of r. The domain of the rule r is the domain of f; the image set of r is the image set of f. The set B is the range of f. With this in mind, we provide the following definition:

Definition 6.3.1. *A (binary) relation $R \subset \Sigma^* \times \Sigma^*$ is said to polynomial bounded if and only if there exists a real polynomial p such that for all x and y in Σ^*, if xRy then $|y| \leq p(|x|)$.*

The above definition also extends to functions. A function f is then polynomial bounded, if and only if there exists a real polynomial p such that for every x in domain of f, $|f(x)| \leq p(|x|)$. We now go on to formally define function problems.

Definition 6.3.2 (Function P). *A polynomial bounded (binary) relation $R \subset \Sigma^* \times \Sigma^*$ is in FP if and only if there exists a deterministic Turing machine such that given an x in Σ^*, if there exists some y in Σ^* where xRy, it outputs such a string y, else it outputs 'no'.*

Definition 6.3.3 (Function NP). *A polynomial bounded (binary) relation $R \subset \Sigma^* \times \Sigma^*$ is in FNP if and only if there exists a non-deterministic Turing machine such that given an x in Σ^*, if there exists some y in Σ^* where xRy, it outputs such a string y, else it outputs 'no'.*

The terminology of function is a bit misleading in the sense that given an x, several y might satisfy xRy. We move to counting problems.

Definition 6.3.4 (Counting Problems). *Given a (binary) relation $R \subset \Sigma^* \times \Sigma^*$, the counting problem $\#R$ associated with R is the function from Σ^* into $\mathbb{N}$ such that for x in Σ^*, $\#R(x) = |\{y \in \Sigma^* : xRy\}|$.*

We now define the class #P.

Definition 6.3.5 (#P). *Given a polynomial bounded (binary) relation $R \subset \Sigma^* \times \Sigma^*$, the counting problem $\#R$ is in #P if and only if there exists a deterministic Turing machine M that can decide R in polynomial time.*

Put differently, the class #P is the class of counting problem associated with languages in NP. The class #P was initially introduced by L. Valiant in 1979. We shall give a natural alternate definition, we define a counting Turing Machine as defined in Valiant's paper (see [15]).

Definition 6.3.6. *A counting Turing machine is a non-deterministic TM with an auxiliary output device that prints in elements of Σ on a special tape the number of accepting computations induced by the input. It has (worst-case) time complexity $f(n)$ if the longest accepting computation induced by the set of all inputs of size n takes $f(n)$ steps (when the TM is regarded as a standard nondeterministic machine with no auxiliary device).*

The class #P is then defined as follows:

Definition 6.3.7. *A function f is in #P if it can be computed by a counting Turing machine with polynomial time complexity.*

We would expect a function f to be #P-Hard, if it is at least as hard as any function in #P. From this perspective, suppose we already know that a function g is #P-Hard, given a function f, if we can reduce the problem of computing the function g to simply computing the function f and derive the value of g from that of f, we should be able to prove hardness of f. However, we need one specific ingredient, in that if we happen to have an answer for f, we can compute the answer for g in polynomial time. Thus, if f can be computed efficiently, we would know that g can be computed efficiently contradicting the fact that g is #P-hard.

The notion of reduction that will be used is one by oracles as described in [15]. An oracle can be thought of a black-box that can answer queries in one time step. An oracle Turing machine is Turing machine with a query tape, an answer tape, and some working tape. To consult the oracle, the Turing machine prints a string on the query tape and, on going into a special query state, an answer is returned in unit time on the answer tape, and a special answer state is entered.

Definition 6.3.8. *An oracle Turing machine is said to be in FP, if and only if for all polynomial bounded oracles, it behaves like a machine in FP.*

If M is a class of oracle TMs, and f a polynomial bounded function, then we denote the class of function that can be computed by oracle TMs in M with oracles for f, by M^f.

Definition 6.3.9. *A polynomial bounded function f is $\#P$-Hard if and only if $\#P \subset FP^f$. A polynomial bounded function f is $\#P$-Complete if and only if $\#P \subset FP^f$ and f belongs to $\#P$.*

In other words, f is #P-Hard if and only if for every function in #P, there exists an oracle Turing machine from FP with oracle for f that can compute it. Given a counting problem #R, to prove that #R is #P-hard, it would be enough to find a problem that is #P-hard, and then reduce this problem to #R. Reduction should be done in such a way that, the construction of the new problem is in polynomial time, and computing the output from the output is in polynomial time. This done, given an oracle TM that has oracle #R, we can compute the other function in polynomial time contradicting Hardness.

It would suffice to mention three counting problems that would be of interest in this thesis. We define the following counting problem:

Definition 6.3.10. *The counting problem $\#3SAT$ takes $< \phi >$ as input where ϕ is a boolean formula in 3-CNF and computes the number of satisfying assignments.*

Proposition 6.3.11. *$\#3SAT$ is $\#P$-Complete.*

Proof. The fact that #3SAT belongs to $\#P$ is rather trivial since deciding the language consists of plugging in the values of the variables and checking if it satisfies the formula. Hardiness follows from the fact that 3CNF is NP-hard. Any polynomial time algorithm used to compute #3CNF may be trivially used to check whether this formula has a satisfying assignment or not. □

One interesting feature is that many decision problems that are *easy*, have hard counting version. In what follows, we identify a problem that is to be used.

Definition 6.3.12. *A boolean formula in CNF is monotone if no clause contains a negation of a variable.*

Definition 6.3.13. *The counting problem #monotone-2SAT takes $< \phi >$ as input where ϕ is a monotone boolean formula in 2-CNF and computes the number of satisfying assignments.*

Proposition 6.3.14. *#monotone-2-SAT is #P-Complete.*

Proof. See [16]. □

Definition 6.3.15. *A boolean formula is in Disjunctive Normal Form (DNF) if it is a disjunction of conjunction clauses. Formally, let $x_1, \cdots x_n$ be boolean variables, a boolean formula ϕ is in DNF-form if ϕ is written as*

$$\phi(x_1, \cdots, x_n) = \bigvee_{i=1}^{m} (y_{i,1} \wedge \cdots \wedge y_{i,k_m}) \tag{6.2}$$

where for each m, k_m is a positive integer, and the literal $y_.$ represents either x_i of $\neg x_i$ for some i.

Definition 6.3.16. *A boolean formula in DNF is monotone if no clause contains a negation of a variable.*

We now prove that the DNF case is also hard.

Definition 6.3.17. *The counting problem #monotone-2DNF takes $< \phi >$ as input where ϕ is a monotone boolean formula in 2-DNF and computes the number of satisfying assignments.*

Lemma 6.3.18. *#monotone-2-DNF is #P-Complete.*

Proof. Let $\phi = \bigwedge_{i=1}^{m}(y_i \vee z_i)$ be a monotone 2-CNF formula. Then the number of satisfying assignment of ϕ equals the number of non-satisfying assignment of $\bigvee_{i=1}^{m}(\neg y_i \wedge \neg z_i)$ which in turn is equal to the number of non-satisfying assignments of $\bigvee_{i=1}^{m}(y_i \wedge z_i)$ by monotonicity and symmetry. Finally, $\bigvee_{i=1}^{m}(y_i \wedge z_i)$ is a monotone 2-DNF, and clearly the number of non-satisfying assignments equals 2^n minus the number of satisfying assignments. $\square$

We turn back to our model.

6.4 The Complexity of Counting Cycles and Fixed Points

We characterize the number of equivalence classes, fixed points and cycles of length two. To this end, we define three counting problems: one for each.

Definition 6.4.1. *The counting problem $\#CYCLE$ takes $< n, G, k >$ as input, where n is a positive integer and (G, k) belongs to $\mathcal{G}_n \times \mathcal{K}_n$, and outputs the cardinality of $CYCLE_n(G, k)$*

Definition 6.4.2. *The counting problem $\#FIX$ takes $< n, G, k >$ as input, where n is a positive integer and (G, k) belongs to $\mathcal{G}_n \times \mathcal{K}_n$, and outputs the cardinality of $\{C \in CYCLE_n(G, k) : |C| = 1\}$*

Definition 6.4.3. *The counting problem $\#2CYCLE$ takes $< n, G, k >$ as input, where n is a positive integer and (G, k) belongs to $\mathcal{G}_n \times \mathcal{K}_n$, and outputs the cardinality of $\{C \in CYCLE_n(G, k) : |C| = 2\}$*

Note that in this setting, given an input $< n, G, k >$, the output of $\#CYCLE$ is equal to the sum of the outputs of both $\#FIX$ and $\#2CYCLE$ when fed with the same input. Referring to the networked coordination game defined in the primary model, $\#FIX$ refers to counting the number of pure Nash equilibria of the networked game. We show the following:

Theorem 6.4.4. *$\#CYCLE$ is $\#$P-Complete.*

Theorem 6.4.5. *$\#FIX$ is $\#$P-Complete.*

Theorem 6.4.6. *$\#2CYCLE$ is $\#$P-Complete.*

One has to be subtle towards what such result entails. This result does not imply that no characterization of the number of cycles is possible whatsoever, but rather that we would be unable to get an arbitrarily refined characterization of that number.

A Note on Encoding: To reconciliate with the language definition provided earlier: given an input $< n, G, k >$, we have that n is an integer and can be easily encoded as a string in $\{0,1\}^*$, the graph G may be represented by its adjacency matrix, and the function k as an array of non-negative integers taking values less than $n+2$.

For technical insight, we may further note that no result in those three implies another, and no two results imply the third (or at least that no deduction may be made simply from the statements above with no additional information whatsoever). In that we mean, if it is hard to count the number of fixed points, counting the number of cycles is not necessarily hard because of set inclusion. As a quick example, consider counting the number of total action configurations, surely this set includes the number of fixed points. However, counting them is trivial given the network size. Similarly, no hardness can directly be deduced by the fact that $\#CYCLE$ outputs the sum of $\#FIX$ and $\#2CYCLE$ when all the counting problems are fed with the same input. To illustrate quickly, consider counting the number of non-fixed point action configurations, this problem is hard since counting the number of fixed points is hard itself, however counting the number of fixed-points and non-fixed-points is again trivial given the size of the network.

We will prove our results as follows, we will restrict our input to only bipartite graphs. Within this restricted space, those three problems share a common ground that will be stated in two lemmas to follow. Mainly, either all of them are hard, or none of them is hard. We then prove the three theorems in one instance by showing that with restricted inputs one of the problems is #P-Hard.

Building on the framework defined in Chapter 4, we begin by this crucial observation.

Lemma 6.4.7. *Let n be a positive integer, (G, k) be in $\mathcal{G}_n^b \times \mathcal{K}_n$ with $G = (\mathcal{I}_n, E)$ and (P_o, P_e) a 2-Partition of $\mathcal{I}_n$ with respect to G. For a in $\mathcal{A}_n$, a is a fixed point of G_k if and only if a is a fixed point of $G_k|_{P_e} G_k|_{P_o}$.*

Proof. It is clear that, if $G_k a = a$, then $G_k|_{P_o} a = G_k|_{P_e} a = a$ and so a is fixed point of $G_k|_{P_e} G_k|_{P_o}$. To show the converse, suppose that $G_k|_{P_e} G_k|_{P_o} a = a$, then $(G_k|_{P_e} G_k|_{P_o} a)_i = a_i$ for all i in $\mathcal{I}_n$. If i is in P_o, it follows that $(G_k|_{P_o} a)_i = a_i$ since $G_k|_{P_e}$ cannot modify a_i, and so $G_k|_{P_o}(a) = a$ since $(G_k|_{P_o} a)_i = a_i$ for i in P_e. It follows that $G_k|_{P_e}(a) = a$. But $(G_k a)_i = (G_k|_{P_o} a)_i$ if i is in P_o, and $(G_k a)_i = (G_k|_{P_e} a)_i$ if i is in P_e. Therefore, $G_k(a) = a$. □

With this in mind, we may proceed to the following curcial lemma.

Lemma 6.4.8. *Let (G, k) be in $\mathcal{G}_n^b \times \mathcal{K}_n$ and let F be equal to $|FIX_n(G, k)|$, then $|2CYCLE_n(G, k)| = F(F-1)/2$ and $|CYCLE_n(G, k)| = F(F-1)/2 + F$.*

Proof. Let (G, k) be in $\mathcal{G}_n^b \times \mathcal{K}_n$ with $G = (\mathcal{I}_n, E)$ and (P_o, P_e) be a 2-Partition of $\mathcal{I}_n$ with respect to G. To prove the result, it would be enough to construct a bijection from the set of non-degenerate cycles in $CYCLE_n(G, k)$ (having a cardinality equal to 2) to $\{(a', a'') \in FIX_n(G, k)^2 : a' \neq a''\}$. To this end, consider two distinct elements a' and a'' in $FIX_n(G, k)$. By Lemma 6.4.7, we know that a' and a'' are fixed points of $G_k|_{P_e} G_k|_{P_o}$. Construct $a_1 = (a'' \upharpoonright P_o, a' \upharpoonright P_e)$ and $a_2 = (a' \upharpoonright P_o, a'' \upharpoonright P_e)$. We get that $\{a_1, a_2\}$ is a non-degenerate cycle in $CYCLE_n(G, k)$. Indeed, $G_k a_1 = a_2$ and $G_k a_2 = a_1$. Finally, clearly the map

$$(a', a'') \mapsto ((a'' \upharpoonright P_o, a' \upharpoonright P_e), (a' \upharpoonright P_o, a'' \upharpoonright P_e))$$

is bijective. The result follows for $|2CYCLE_n(G, k)|$ from the fact that a cycle in $CYCLE_n(G, k)$ is an unordered pair of action configurations. The result for $|CYCLE_n(G, k)|$ is then immediate. □

This lemma states that we need only prove the result for counting fixed-points over bipartite graphs. We begin by a technical lemma.

Lemma 6.4.9. *Consider a graph (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where $G = (\mathcal{I}_n, E)$. Suppose that there exists P_s and P_c disjoint subsets of $\mathcal{I}_n$ such that k is equal to 2 on $P_s \cup P_c$ and*

- *The cardinality of P_s is exactly equal to 3.*
- *The cardinality of P_c is greater than or equal to 3.*
- *For all (s, c) in $P_s \times P_c$, we have $sc \in E$.*
- *For each s in P_s there exists no node j in $\mathcal{I}_n \backslash P_c$ such that $sj \in E$.*
- *Every node in P_c has degree less than or equal to 4.*

If a in $\mathcal{A}_n$ is a fixed point of G_k, then a can take only one value over $P_s \cup P_c$.

Proof. Let a be a fixed point of G_k, and suppose there are at least two nodes i and j in P_c such that, $a_i = a_j = \mathbb{B}$. Since a is a fixed point, a is equal to $\mathbb{B}$ on P_s, it then follows that a is equal to $\mathbb{B}$ on P_c. Similarly, suppose there is at most one node i in P_c such that a_i is $\mathbb{B}$. Since a is a fixed point, a is equal to $\mathbb{W}$ on P_s, it then follows that $a_i = \mathbb{W}$, contradicting the assumption. □

Let us define #bipartite-FIX to be the counting problem #FIX while restricting an input $< n, G, k >$ to have G being a bipartite graph in $\mathcal{G}_n^b$. We get the following theorem:

Theorem 6.4.10. *The counting problem #bipartite-FIX is #P-Complete.*

Proof. Clearly #bipartite-FIX is in #P, an action configuration can be easily verified to be a fixed-point in polynomial time. Let ϕ be a monotone 2-DNF formula of m clauses and n literals $x_1, \cdots, x_n$. Namely,

$$\phi(x_1, \cdots, x_n) = \bigvee_{c=1}^{m} (x_{y_c} \wedge x_{z_c}), \tag{6.3}$$

where y and z are maps from $\{1, \cdots, m\}$ into $\{1, \cdots, n\}$. Without any loss of generality we will assume that all literals appear in the formula, otherwise we can reduce it to formula having a fewer number of literals. Moreover, if a clause consists on only one literal x, we will write it as $x \wedge x$.

We construct a graph $G(\mathcal{I}_{3(n+2m+m+1)}, E)$ of size $3(n + 2m + m + 1)$ as follows. We consider $\mathcal{I}_{3(n+2m+m+1)}$ and label the players as

- s_p^1, s_p^2, s_p^3 for $p \in \{1, \cdots, n\}$.
- $y_c^1, y_c^2, y_c^3, z_c^1, z_c^2, z_c^3$ for $c \in \{1, \cdots, m\}$.
- b_c^1, b_c^2, b_c^3 for $c \in \{1, \cdots, m\}$.
- d^1, d^2, d^3.

We now let the edge set E contain the following edges:

- $b_c^l y_c^l$ and $b_c^l z_c^l$ for all $l \in \{1, 2, 3\}$ and $c \in \{1, \cdots, m\}$.
- $d^l b_c^l$ for all $l \in \{1, 2, 3\}$ and $c \in \{1, \cdots, m\}$.
- $y_c^l y_{c'}^{l'}$ if $y_c = y_{c'}$ for c, c' in $\{1, \cdots, m\}$ and l, l' in $\{1, 2, 3\}$, $c \neq c'$ or $l \neq l'$.
- $z_c^l z_{c'}^{l'}$ if $z_c = z_{c'}$ for c, c' in $\{1, \cdots, m\}$ and l, l' in $\{1, 2, 3\}$, $c \neq c'$ or $l \neq l'$..
- $y_c^l z_{c'}^{l'}$ if $y_c = z_{c'}$ for c, c' in $\{1, \cdots, m\}$ and l, l' in $\{1, 2, 3\}$.
- $s_p^l y_c^{l'}$ if $y_c = p$ for p in $\{1, \cdots, n\}$, c in $\{1, \cdots, m\}$ and l, l' in $\{1, 2, 3\}$.

We define k in $\mathcal{K}_{3(n+2m+m+1)}$ such that k is equal to 2 everywhere on $\mathcal{I}_{3(n+2m+m+1)}$ except at d^1,d^2 and d^3 where it is equal to 1.

Let f be an assignment for ϕ. Define A_f to be the subset of $FIX_{3(n+2m+m+1)}(G, k)$ such that for a in A_f,

- $a(y_c^l) = \mathbb{B}$ iff $f(y_c) = 1$, for c in $\{1, \cdots, m\}$ and l in $\{1, 2, 3\}$,
- $a(z_c^l) = \mathbb{B}$ iff $f(z_c) = 1$, for c in $\{1, \cdots, m\}$ and l in $\{1, 2, 3\}$.

We claim three things. First, for every satisfying assignment sat of ϕ, the cardinality of A_{sat} is equal to 1. Second, If 0 is the all zero assignment of ϕ, the cardinality of A_0 is equal to 1. Third, for every non-satisfying assignment $nsat$ of ϕ different than the all zero assignment, the cardinality of A_{nsat} is equal to 8.

Let sat be a satisfying assignment of ϕ, namely a mapping sat from $\{1, \cdots, n\}$ into $\{0, 1\}$ such that $\phi(sat) = 1$. For a in A_{sat}, all nodes corresponding to the same literals have the same action, and by lemma 6.4.9, since a is a fixed point, all s_p^l have the same color as the nodes connected to them. Since sat is a satisfying assignment, at least one clause is satisfied. Let that clause be c_{sat}. It then follows that since a is a fixed point, $b_{c_{sat}}^l$ is $\mathbb{B}$, and so d^l is $\mathbb{B}$. All other actions are then deterministically set. Therefore, $A_{sat} = \{a\}$.

In the case of the all zero assignment 0, let a be an element of A_0. Again, all nodes corresponding to the same literals have the same action, and by Lemma 6.4.9, all s_p^l have the same color as the nodes connected to them. The assignment 0 is non-satisfying, therefore d^l is $\mathbb{W}$. It then follows the all nodes in the graph are $\mathbb{W}$. Therefore $A_0 = \{a\}$.

Finally, let $nsat$ be a non-satisfying assignment of ϕ different than the all zero assignment, and let a be an element of A_{nsat}. First we have that all s_p^l have the same color as the nodes connected to them. Since $nsat$ is not the all zero assignment, there exists c such that either y_c^l is $\mathbb{B}$ or z_c^l is $\mathbb{B}$. Pick l in $\{1, 2, 3\}$. Suppose that a^l is $\mathbb{B}$ then b_c^l is $\mathbb{B}$ and the color of all $b_{c'}^l$ with $c \neq c'$ are deterministically set. Suppose that a^l is $\mathbb{W}$ then $b_{c'}^l$ is $\mathbb{W}$ for all c'. Therefore, $|A_{nsat}| = 8$.

For assignments f and f' for ϕ, clearly if $f \neq f'$ then $A_f \cap A_{f'} = \emptyset$. It also follows from Lemma 6.4.9 that for any fixed point a of G_k, there exists an assignment f of ϕ such that $a \in A_f$. Indeed all the nodes corresponding to the same literal share the same color. Then consider a set of n nodes corresponding to the different literals (such a set exists since we assumed that all literals appear in the formula), the coloring of those nodes translates to an assignment f of ϕ such that A_f contains the fixed point considered.

Let $\#sat$ and $\#nsat$ be respectively the number of satisfying and non-satisfying assignments of ϕ. Let F be the cardinality of $FIX_{3(n+2m+m+1)}(G,k)$, then

$$\#sat + \#nsat = 2^n \quad \text{and} \quad \#sat + 8(\#nsat - 1) + 1 = F.$$

The graph can be constructed in polynomial time, the system of equation can also be solved in polynomial time. □

If we define #bipartite-CYCLE and #bipartite-2CYCLE to be respectively the counting problems #CYCLE and #2CYCLE while restricting an input $< n, G, k >$ to have G a bipartite graph in G_n^b, we arrive to the following corollaries.

Corollary 6.4.11. *#bipartite-CYCLE is #P-Complete.*

Proof. Combine Lemma 6.4.8 and Theorem 6.4.10. □

Corollary 6.4.12. *#bipartite-2CYCLE is #P-Complete.*

Proof. Combine Lemma 6.4.8 and Theorem 6.4.10. □

The results stated in theorem 6.4.4, 6.4.5 and 6.4.6 hold then by inclusion. We can even claim stronger statements, notice that we only used k equal to 1 or 2. Even more, we could make the construction by restricting k to be only equal to 2 everywhere only by doubling the number of nodes in the graph. In some sense, the complexity is not truly coming from the heterogeneity of the threshold, but rather from the threshold rule and the network complexity. To formalize the idea, let us denote by #2-bipartite-FIX, #2-bipartite-CYCLE and #2-bipartite-2CYCLE respectively the counting problems #bipartite-FIX, #bipartite-CYCLE, and #bipartite-2CYCLE while restricting an input $< n, G, k >$ to having both G a bipartite graph in G_n^b and k equal to 2 on all nodes. We get:

Theorem 6.4.13. *The counting problems #2-bipartite-FIX is #P-Complete.*

Proof. The proof follows exactly from that of Theorem 6.4.10, however we will first make two copies of the obtained graph and threshold distribution. We then pair the nodes having a threshold equal to 1 and connect each pair with an edge. We finally contract the edges in each obtained pair: each pair becomes a single node. We finally impose a threshold equal to 2 on those new formed nodes. □

Corollary 6.4.14. *#2-bipartite-CYCLE is #P-Complete.*

Proof. Combine Lemma 6.4.8 and Theorem 6.4.13. □

Corollary 6.4.15. *#2-bipartite-2CYCLE is #P-Complete.*

Proof. Combine Lemma 6.4.8 and Theorem 6.4.13. □

General graph structures incur a good amount of complexity, however ordered graph structure can fairly tractable. In what follows we revisit the case of the complete graphs and show that counting over complete graphs is in FP.

6.5 Counting on Complete Graphs

We show that counting cycles, fixed points and non-degenerate cycles are in FP when the graph structure is restricted to complete graphs. We shall not provide algorithm, we would settle on providing enough lemmas and insight to be able to fill up any gap leading to a construction of an algorithm.

Given a pair (G, k) in $\mathcal{G}_n \times \mathcal{K}_n$ where G is a complete graph, recall that an action configuration a in $\mathcal{A}_n$ belongs to $\mathcal{B}_k$ if and only if for every player i in $\mathcal{I}_n$, if player i plays $\mathbb{B}$ in a, then each player j with $k_j < k_i$ plays $\mathbb{B}$ in a. Moreover, recall that $a \in \mathcal{B}_k$ is said to be at level L if and only if some node i in $\mathcal{I}_n$ is playing $\mathbb{B}$ in a and every node j having $k_j > L$ is playing $\mathbb{W}$ in a. If a is $\mathbb{W}$ everywhere, it is said that a is at level -1. Furthermore, from Proposition 5.3.2, we know that each action configuration a in $\mathcal{B}_k$ is at some level L in $\{-1\} \cup \mathbb{N}$. Therefore, given a pair (G, k), if we define A_L to be the collection of action configuration in $\mathcal{B}_k$ at level L, for a finite subset $\mathcal{L}$ in $\{-1\} \cup \mathbb{N}$, the set $A_{\mathcal{L}} = \{A_L : L \in \mathcal{L}\}$ partitions $\mathcal{B}_k$. Moreover, since we have n players, $\mathcal{L}$ can be chosen in such a way that $|A_{\mathcal{L}}| = O(n)$. Indeed, each threshold value would correspond to one level L and -1 is appended to the set.

We shall consider such a collection $\mathcal{L}$ where A_L is non-empty for each L in $\mathcal{L}$. First, we note that since $G_k(a) \in \mathcal{B}_k$, then the fixed points and non-degenerate cycles consist only of action configurations in $\mathcal{B}_k$. It follows by definition that each fixed point is at only one level, and each non-degenerate cycle contains action configurations at only one level. To show that the problem is in P, we first show that each level can be the level of at most one fixed point, and each level can contain action configurations of at most one non-degenerate cycle (up to isomorphism). We shall formalize those facts.

Proposition 6.5.1. *A level L in $\mathcal{L}$ can be the level of at most one fixed-point of G_k.*

Proof. If $L = -1$ the result trivially follows. We assume that $L \neq -1$. Suppose that a and b are two fixed-points of G_k that are at level L, then necessarily each node i in $\mathcal{I}_n$ with threshold $k_i \leq L$ plays $\mathbb{B}$ in both a and b. Furthermore, each node i in $\mathcal{I}_n$ with threshold $k_i > L$ plays $\mathbb{W}$ by definition. It follows that $a = b$. □

Proposition 6.5.2. *If a level L in $\mathcal{L}$ is the level of some action configuration in a non-degenerate cycle, it is the level of action configurations of $\binom{m}{m/2}/2$ non-degenerate cycle of G_k where $m = k^{-1}(L)$.*

Proof. First, each non-degenerate cycle contains action configurations at levels different than -1. All nodes playing $\mathbb{W}$ can only be a fixed-point. This said, we suppose L is different than -1. It is simple to see that every non-degenerate cycle contains action configurations in at most one level. Furthermore, assume that L is the level of some action configuration in a non-degenerate cycle. A non-degenerate cycle can exist at level L only if $k^{-1}(L)$ is even by Proposition 4.2.5. In that case, every non-degenerate cycles contains action configurations that are $\mathbb{B}$ on half of the nodes having threshold L and $\mathbb{W}$ on the other half. Furthermore, each non-degenerate cycle would consist of two action configuration such that $a'_i \neq a_i$ for all i. If $m = k^{-1}(L)$, we would have $\binom{m}{m/2}/2$ such non-degenerate cycles. □

Define $L_{max} = \max \mathcal{L}$. For $L \neq L_{max}$ in $\mathcal{L}$, define $L+$ to be the smallest L' in $\mathcal{L}$ greater than L. Furthermore, for every L in $\mathcal{L}$, define S_L to be the number of nodes having threshold less than or equal to L.

Proposition 6.5.3. *The level L_{max} is a level of a fixed-point if and only if $L_{max} < n$.*

Proof. The only candidate to be a fixed-point at level L_{max} is the action configuration where all nodes are playing $\mathbb{B}$. This configuration can only be sustained if $L_{max} < n$. Conversely, if $L_{max} < n$, all nodes playing $\mathbb{B}$ is clearly a fixed-point. $\square$

Proposition 6.5.4. *A level $L \neq L_{max}$ in $\mathcal{L}$ is a level of a fixed point if and only if $L+ > S_L$ and $S_L - 1 \geq L$.*

Proof. If $L+ > S_L$, then if a is at level L, $G_k a$ is at a level less than $L+$. If $S_L - 1 \geq L$, if $a \in \mathcal{B}_k$ is such that all players at level L are playing $\mathbb{B}$, then $G_k a$ is necessarily at a level greater than or equal to L. Having those two conditions imply that if $a \in \mathcal{B}_k$ is at level L in such a way that all players at level L are playing $\mathbb{B}$, then $G_k a = a$. The converse is straightforward. $\square$

Proposition 6.5.5. *A level L is a level of action configurations in a non-degenerate cycle if and only if $m = |k^{-1}(L)|$ is even, and $S_L - m/2 = L - 1$.*

Proof. The proof follows from Proposition 4.2.5. $\square$

To end the section, checking if a level L is a level for either fixed points or action configurations of non-degenerate cycles may be done in $O(n)$. Using the propositions presented, we can in $O(n)$ steps go over each level in $\mathcal{L}$ and count the number of fixed-points and non-degenerate action configurations residing in each.

Restricting the counting to graph structure being cycle graphs and trees with bounded degrees can be also shown to be in FP, though we will not provide any details.

6.6 On Reachability and Counting Predecessors

A different question of interest is to decide whether given a graph structure G, a type distribution k and some action configuration a, the action configuration a is reachable from some configuration b. We define the following:

Definition 6.6.1. *The language PRED consists of all 4-tuples $< n, G, k, a >$, where n is a positive integer, (G, k, a) belongs to $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ with $G_k(a)^{-1} \neq \emptyset$.*

We get the following result:

Theorem 6.6.2. *PRED is NP-Complete.*

Proof. First, clearly PRED is in NP. We now perform a reduction from 3SAT. Let ϕ be a 3-CNF formula of m clauses and n literals $x_1, \cdots, x_n$. Namely,

$$\phi(x_1, \cdots, x_n) = \bigwedge_{c=1}^{m} (y_c \vee w_c \vee z_c), \tag{6.4}$$

We construct an undirected graph $G(\mathcal{I}_{4n+m+1}, E)$ as follows. We consider $\mathcal{I}_{4n+m}$ and label the players as

- v_p, v'_p, o_p, t_p for $1 \leq p \leq n$.
- s_c for $1 \leq c \leq m$.
- u.

We let E contains the following undirected edges:

- o_pv_p, $o_pv'_p$, t_pv_p, and $t_pv'_p$ for all p.
- s_cv_p if and only if $x_p \in \{y_c, w_c, z_c\}$ and $s_cv'_p$ if and only if $\neg x_p \in \{y_c, w_c, z_c\}$ for all p and c.
- uv_p and uv'_p for all p.

We define k in $\mathcal{K}_{4n+m+1}$ to be equal to 1 everywhere on $\mathcal{I}_{4n+m+1}$ except at $t_1, \cdots, t_m$ where it is equal to 2. Finally, construct a in $\mathcal{A}_{4n+m+1}$ such that a is $\mathbb{B}$ on $\mathcal{I}_{4n+m+1}$ except at $t_1, \cdots, t_m$ where it $\mathbb{W}$. We claim that there exists b in $\mathcal{A}_{4n+m}$ such that $a = G_kb$ if and only if ϕ is satisfiable. To see that, suppose such a b exists. We have that v_p and v'_p represent the variable x_n and its negation $\neg x_n$ respectively. The node o_p being $\mathbb{B}$ enforces either v_p to be $\mathbb{B}$ or v'_p to be $\mathbb{B}$, t_p being $\mathbb{W}$ enforces either v_p to be $\mathbb{W}$ or v'_p to be $\mathbb{W}$. It follows that v_p and v'_p are of opposite colors in b.

Finally, s_c being $\mathbb{B}$ enforces that clause c is satisfied. The node u is only needed to ensure connectedness of the graph. To prove the converse, suppose ϕ has a satisfying assignment, let *sat* be such an assignment. Construct b as follows: make b equal to $\mathbb{B}$ everywhere on $\mathcal{I}_{4n+m+1}$ except on v_p and v'_p for all p. Finally, set $b_{v_p} = \mathbb{B}$ if and only if $x_p = 1$ and $b_{v'_p} = \mathbb{B}$ if and only if $x_p = 0$ for all p. Finally, the construction of the graph is done in polynomial time. $\square$

Given a graph structure G, a type distribution k and a configuration a, suppose we want to compute the number of configurations b from which a can be reached by applying G_k only once on b. We define

Definition 6.6.3. *The counting problem #PRED takes* $< n, G, k, a >$ *as input, where* n *is a positive integer and* (G, k, a) *and outputs the cardinality of* $G_k^{-1}(a)$.

As a corollary from the hardness of PRED, we get:

Corollary 6.6.4. *#PRED is #P-Complete.*

However, suppose that we restrict the counting to only the action configurations that are reachable from some action configuration. Specifically, we restrict the counting to only the elements in $PRED$. From this perspective, we are computing the 'fan-in' of given configuration.

Definition 6.6.5. *The counting problem #reachable-PRED takes* $< n, G, k, a >$ *as input, where* n *is a positive integer,* (G, k, a) *belongs to* $\mathcal{G}_n \times \mathcal{K}_n \times \mathcal{A}_n$ *and* $G_k(a)^{-1} \neq \emptyset$ *then outputs the cardinality of* $G_k^{-1}(a)$.

We get the following result:

Theorem 6.6.6. *#reachable-PRED is #P-Complete.*

Proof. First, the fact that #reachable-PRED belongs to #P is clear. We now perform a reduction from #monotone-2-CNF. Let ϕ be a Monotone-2-CNF formula of m clauses and n literals $x_1, \cdots, x_n$. Namely,

$$\phi(x_1, \cdots, x_n) = \bigwedge_{c=1}^{m} (y_c \vee z_c), \tag{6.5}$$

We construct a graph $G(\mathcal{I}_{n+m+1}, E)$ as follows. We consider the set $\mathcal{I}_{n+m+1}$ and label the nodes as follows $v_1, \cdots, v_n$, $u_1, \cdots u_m$ and d. Construct the E in such a way that, $u_c v_p$ in E if and only if x_p appears in clause c. Finally we let dv_p belong to E for every p. Define k in $\mathcal{K}_{n+m+1}$ to equal to 1 everywhere on $\mathcal{I}_{n+m+1}$ except at u_l for all l where it is equal to 2.

Let a be the action configuration in $\mathcal{A}_{n+m+1}$ such that a is equal to $\mathbb{B}$ everywhere. Clearly a is reachable in (G, k) since it is a fixed-point. We claim that the number of configurations preceding a, i.e. $|G_k^{-1}a|$ is equal to the number of satisfying assignments for ϕ. To show that we set up a bijection from $G_k^{-1}a$ into the set of satisfying assignment. Let b be any action configuration in $G_k^{-1}a$, then necessarily b is $\mathbb{B}$ on u_l for all l and on d. Any coloring configuration on the v_p and v'_p nodes that would induce an action configuration in $G_k^{-1}a$ is actually a satisfying assignment, and any satisfying assignment translates as a coloring on the v_p and v'_p nodes that yields an action configuration in $G_k^{-1}a$. Finally, clearly #reachable-PRED is in #P. $\square$

We transition to the resilience analysis context in the next chapter.

Chapter 7

Resilience of Networks

In this section, we revert back to the primary model (see Section 2.1)where we consider *types* instead of *thresholds*, namely $\mathcal{Q}_n$ instead of $\mathcal{K}_n$. All the needed definitions in this thesis including K_n naturally extend to the set Q_n. Mainly, for $G(I_n, E)$ in $\mathcal{G}_n$ and q in $\mathcal{Q}_n$, we denote by G_q the map from $\mathcal{A}_n$ into $\mathcal{A}_n$ such that for player i, $(G_k a)_i = \mathbb{B}$ if and only if at most $q_i d_i$ players are in $a^{-1}(\mathbb{B}) \cap \mathcal{N}_i$.

7.1 The Resilience Measure

We consider the following resilience problem. Define $||.||_1$ to be the map from $\mathcal{Q}_n$ into $\mathbb{R}$ such that, for q in $\mathcal{Q}_n$,

$$||q||_1 = \sum_{i \in \mathcal{I}_n} q_i.$$

We restrict the analysis in the thesis to the map $||.||_1$. Let K be a positive integer, we denote by $\mathcal{A}_n^K$ the subset of $\mathcal{A}_n$ such that, a is in $\mathcal{A}_n^K$ if and only if the cardinality of $a^{-1}(\mathbb{B})$ is at most K. We denote respectively by $\mathbb{W}^n$ and $\mathbb{B}^n$ the (constant) action configurations in $\mathcal{A}_n$ mapping each player in $\mathcal{I}_n$ into $\mathbb{W}$ and $\mathbb{B}$ respectively. Recall that for a and b in $\mathcal{A}_n$, we have $b\mathcal{R}_{G_q}a$ if and only if $b = G_q^m a$ for some non-negative integer m (See Chapter 3). Given a graph G in G_n, we define $\mathcal{Q}_n^{G,K}$ to be the subset of $\mathcal{Q}_n$ such that for every q in $\mathcal{Q}_n^{G,K}$ and a in A_n^K, we have $\mathbb{W}^n\mathcal{R}_{G_q}a$. We define

$\mu_n^K(G)$ to be the resilience measure of a graph G with respect to K deviations to be

$$\mu_n^K(G) = \inf\{\ ||q||_1 : q \in \mathcal{Q}_n^{G,K}\ \}.$$

Note that without any loss of generality, we may assume that for any q in $\mathcal{Q}_n^{G,K}$, q_i is of the form m/d_i for $0 \le m \le d_i$.

We explain the problem formulation. Given a graph structure G and a positive integer K, we suppose that at most K players in the network start playing action $\mathbb{B}$. The goal is to allocate a type distribution q over the players, so that the dynamics depicted in Proposition 2.1.1 lead the agents to play action $\mathbb{W}$ at the limit. From this perspective, the measure μ captures the minimal cost of type investment required to recover the network G from a perturbation of magnitude K. In this sense, the lower the resilience measure is for a graph G, the more robust G is against perturbations, in that we mean the less costly it is to allocate types to have G recover. We prove some bounds.

7.2 On Lower Bounds

We prove lower bounds on the resilience measure.

Theorem 7.2.1. *The resilience measure μ_n^K is greater than or equal to 1 for every positive integer $K \le n$.*

Proof. Without any loss of generality, we may assume that $K = 1$. Let $G(\mathcal{I}_n, E)$ be a graph in $\mathcal{G}_n$, and set $d_{max} = n - 1 - m = \max_{i \in \mathcal{I}_n} d_i$, for some non-negative integer $m < n - 1$. Let q in $\mathcal{Q}_n$ be given, and let k be the number of zero coordinates in q, namely the cardinality of $q^{-1}(0)$. If $k = n$, then at least two nodes i and j with zero q_i and q_j satisfy $ij \in E$, and it would follow that $q \notin \mathcal{Q}_n^{G,1}$. We then have $k < n$.

Every player i with $q_i = 0$ will add $1/d_j$ to every q_j with j in $\mathcal{N}_i$. If $m + 1 \ge k$ then $n - k \ge d_{max}$ and since $q^{-1}(0)$ has cardinality k, we get

$$||q||_1 \ge \frac{n-k}{d_{max}} \ge 1.$$

Suppose that $m+1 < k$, then there exists at least one player that is connected to $n-m-1$ players, or put differently, that leaves m players not connected to it. Suppose that player i with $d_i = d_{max}$ has $q_i = 0$, it would follow that $||q||_1 \geq d_{max}/d_{max} \geq 1$. So assume that at least some player i with $d_i = d_{max}$ has $q_i > 0$. Then the players in $q^{-1}(0)$ can be connected to at most $1+m$ distinct players, one of them being player i, and so

$$||q||_1 \geq \frac{k}{d_{max}} + \frac{n-(m+1)}{d_{max}} \geq 1.$$

□

We show that the bound is tight.

Proposition 7.2.2. *This bound is achieved by the star graph S_n for all n and K. The star graph S_n is the unique optimal solution for $K > 1$.*

Proof. To show that the bound is achieved by the star network, allocate types on the graph such that the node with degree $n-1$ has a type of 1 and the rest a type of 0. We now prove uniqueness for $K > 1$.

Given n, suppose $d_{max} = n-m-1$, and let k be the number of zeros in the graph. If $m+1 > k$, we are done since

$$||q||_1 \geq \frac{n-k}{d_{max}} > 1. \tag{7.1}$$

So suppose $m+1 \leq k$, and suppose we can minimally cover those k nodes with p nodes. That is, let S be a subset of $V(G)$ having the smallest cardinality, such that each node of those k nodes is connected to a node in S. We suppose that $|S| = p$. Then $1 \leq p \leq m+1$. If $p = 1$, we have the case of the star network. So suppose $p > 1$, then

$$||q||_1 \geq \frac{k}{d_{max}} + \frac{n-k-p}{d_{max}} > \frac{n-p}{d_{max}}.$$

Therefore, if $p < m+1$, we have that $||q||_1 > 1$. We then suppose $p = m+1$. Furthermore, we suppose two of the nodes in S are connected by an edge. Since

$K = 2$, one of those nodes (call it i) has a type equal to at least $2/d_i$, and so:

$$||q||_1 \geq \frac{k}{d_{max}} + \frac{1}{d_{max}} + \frac{n-k}{d_{max}} > 1.$$

Finally, suppose none of those two nodes are connected, then each can have a maximum degree of $n - 1 - m - 1$, since each node is not connected to the m others and at least one of the zeros is not connected to it. It follows that:

$$||q||_1 \geq \frac{k}{d_{max} - 1} + \frac{n-k}{d_{max}} > 1.$$

□

For $K = 1$, the complete graph is also an optimal solution.

7.3 On Upper Bounds

Theorem 7.3.1. *The resilience measure μ_n^K is less than or equal to $n/2$ for every positive integer $K \leq n$.*

Proof. Without any loss of generality, we may assume that $K = n$. In this case, the players start by playing the action configuration $\mathbb{B}^n$, and we need all the players to play $\mathbb{W}$ at the limit. Impose a strict order relation $<$ on $\mathcal{I}_n$, such that for every i and j in $\mathcal{I}_n$,

$$i < j \quad \text{if} \quad d_i < d_j. \tag{7.2}$$

It is to note that the statement in (7.2) is not an *only if* statement. The case where $d_i = d_j$ is taken care of by the fact that $<$ is a strict order. We construct q as follows: for i in $\mathcal{I}_n$, set

$$q_i = \sum_{j \in \mathcal{N}_i : j < i} d_i^{-1}.$$

We are iterating over all edges i, j, and adding d_k^{-1} to q_k of player k in $\{i, j\}$ that has the highest degree, or if the degrees are equal that is the tie breaker set by the order relation $<$.

Then, the type distribution q is in $\mathcal{Q}_n^{G,n}$. To show that, we set up an order preserving bijection r^{-1} from $(\mathcal{I}_n, <)$ into $(\{1, \cdots, n\}, >)$, and we refer to player $r(k)$ as simply player k, for k in $\{1, \cdots, n\}$. So player 1 refers to the 'largest' player. We claim that player k will be playing $\mathbb{W}$ after applying G_q^k. We prove this by induction. Player 1 will have necessarily have $q_{r(1)} = 1$, and so will necessarily play $\mathbb{W}$ when we apply G_q. Suppose the statement is true for player k, we show that the statement is true for player $k+1$. After G_q^k, all players k' with $k' \leq k$ are playing $\mathbb{W}$. Assume node $k+1$ has degree $d_{r(k+1)}$, and suppose it is connected to m players 'smaller' than it, then it has $q_{r(k+1)} = md_{k+1}^{-1}$, and so it needs more than m neighbors $\mathbb{B}$ to play $\mathbb{B}$, but all the players that are 'larger' than it are playing $\mathbb{W}$, so player $k+1$ will play $\mathbb{W}$ when G_k is applied one more time.

Finally, $||q||_1 \leq n/2$. To prove that, each node i has degree d_i, and so can contribute no more than $d_i \; d_i^{-1} = 1$ to $||q||_1$. If we give each player i a type $q_i = 1$, each edge is then counted twice in the summation, then

$$\begin{aligned} n = \sum_{i \in \mathcal{I}_n} \sum_{j \in \mathcal{N}_i} \frac{1}{d_i} &= \sum_{ij \in E} \frac{1}{d_i} + \frac{1}{d_j} \\ &= \sum_{ij \in E} \frac{1}{d_i} \wedge \frac{1}{d_j} + \frac{1}{d_i} \vee \frac{1}{d_j}. \end{aligned} \tag{7.3}$$

But by construction, we know that

$$||q||_1 = \sum_{ij \in E} \frac{1}{d_i} \wedge \frac{1}{d_j}. \tag{7.4}$$

The result follows since $d_i^{-1} \wedge d_j^{-1} \leq d_i^{-1} \vee d_j^{-1}$. □

We show that the bound is tight for large K.

Proposition 7.3.2. *This bound is achieved by the 2-regular connected graph R_n for all n and $K \geq \lceil n/2 \rceil$.*

Proof. It would be enough to show that the bound is achieved for $K = \lceil n/2 \rceil$ by the 2-regular connected graph.

Let us consider the case where n is even. We begin by claiming, that for any distinct i and j in $\mathcal{I}_n$, if both q_i and q_j are 0, then for any (vertex) path P from i to j there exists a node k in $\mathcal{I}_n$ (distinct from i and j) such that k lies on the (vertex) path and k has a type equal to 1. To prove that, we pick two nodes i and j, and consider a path $(i, i_1, \cdots, i_m, j)$ from i to j. Since i and j are distinct, then $m < n-2$. We now suppose there are no nodes having a type equal 1 along the path, then all the types along the path are less than 1. Consider some a in $\mathcal{A}_n^K$ such that $a_i, a_{i_2}, \cdots, a_{i_m}$ are all $\mathbb{B}$, that is possible since $m/2+1 < n/2$, applying G_q once on a yields $a_{i_1}, \cdots, a_{i_{m-1}}, a_j$ are all $\mathbb{B}$, applying it one more time yields $a_i, a_{i_2}, \cdots, a_{i_m}$ are all $\mathbb{B}$ again. Therefore, there exists a in $\mathcal{A}_n^K$ such that a is not in the same equivalence class as $\mathbb{B}^n$, i.e. $(\mathbb{B}^n, a) \notin \mathcal{R}_{G_q}$.

The second claim is that we need at least one node in the graph having a type equal to 1, otherwise not all nodes will play $\mathbb{W}$ at the limit. To see that, suppose no such node exists. We construct a in $\mathcal{A}_n^K$ such that no two neighboring players have the same action. Applying G_q once on a makes at least all the players that were playing $\mathbb{W}$ play $\mathbb{B}$, and applying it one more time makes the initial players that were $\mathbb{B}$ play $\mathbb{B}$ again. Therefore, a does not belong to $\mathcal{A}_n^K$.

We cannot have more than $n/2$ nodes with type 0 in the graph, otherwise necessarily two nodes with type 0 will be connected. Then, suppose we have k nodes having type 0, we get at least k nodes having type 1 and the rest is $1/2$. If we sum the types, we get $n/2$.

For the case where n is odd, following a similar argument, we establish that we should have at least one node with a type equal to 1 in the network. Then, for any two disjoint players, every (vertex) path connecting the two players should contain at least one node having a type equal to 1. □

7.4 Resilience of Cycle Graphs and Complete Graphs

We derive the resilience of path graphs, cycle graphs and complete graphs.

Proposition 7.4.1. *The path graph P_n of size n has a resilience measure $\mu_n^K(R_n)$*

equal to $(n - \lfloor \frac{n-1}{2K+1} \rfloor)/2$ *for* $K < \lceil n/2 \rceil$.

Proof. The nodes at end points of the path have necessarily a type equal to 0. Suppose no node has a type equal to 1, then types are either equal to 0 or equal to 1/2. Given that K players start playing $\mathbb{B}$, each two players with type equal to 0 should be separated by at least $2K$ players each with type equal to 1/2. If k is the number of players with type equal to 0, then $(n-k)/(k-1) \geq 2K$. Maximizing $(n-k)/(k-1)$ over the number of type 0 nodes, yields $k = \lfloor \frac{n-1}{2K+1} \rfloor$. Suppose a node with type 1 exists, we may split the path into two smaller paths, and it can be checked that is yield only a suboptimal type distribution. □

Proposition 7.4.2. *The cycle graph* R_n *of size* n *has a resilience measure* $\mu_n^K(R_n)$ *equal to* $(n - \lfloor \frac{n}{2K+1} \rfloor)/2$ *for* $K < \lceil n/2 \rceil$.

Proof. We first show that there exists an optimal allocation of types that is nowhere equal to 1 on $\mathcal{I}_n$. Suppose some node has a type equal to 1, then we can delete that node and obtain a path. The optimal allocation in that path is one with no node having a type equal to 1. Therefore, we can only have one node having type equal to 1 if ever. If this node is connected to a node with type equal to 0, we can replace the two types by 1/2 while keeping a type distribution in $\mathcal{Q}_n^K$. If this node is connected to two nodes with types equal to 1/2, we can swap the types of one of the neighbors with the initial node while keeping a type distribution in $\mathcal{Q}_n^K$. We can keep 'moving' the type equal to 1 till its corresponding node is connected to a node having a type of 0. We then apply the previous argument.

This said, an optimal allocation need not create a node having type equal to 1. Let us assume we have such an allocation, and let us determine the maximum number of type 0 nodes we can include. Repeating a similar argument to that of the previous proof yields $k = \lfloor \frac{n}{2K+1} \rfloor$. The result follows. □

Proposition 7.4.3. *The complete graph* K_n *of size* n *has a resilience measure* $\mu_n^K(K_n)$ *equal to* $\frac{K(K-1)/2+K(n-K)}{n-1}$ *for all* K.

Proof. We consider the type distribution $\hat{q}$ such that $\hat{q}$ equals K/d on exactly $n-K$ players, and is injective when restricted to the remaining players taking values in

$\{0, 1/d, \cdots, (K-1)/d\}$. We claim that $\hat{q}$ belongs to Q_n^K. To see that, we argue as follows. Each node is initially connected to either K or $K-1$ neighboring nodes playing $\mathbb{B}$. Therefore, the type distribution q equal to K/d everywhere is in $\mathcal{Q}_n^K$. This distribution is however not optimal. We can allow K nodes in K_n to have types equal to $(K-1)/d$, while the rest have types of K/d and keep the distribution in $\mathcal{Q}_n^K$. Let that distribution be q'. Given any initial configuration in $\mathcal{A}_n^K$, after one application of $G_{q'}$, only K of the nodes can play $\mathbb{B}$. The $n-K$ nodes having types equal to K/d will always play $\mathbb{W}$, therefore we can delete them from the network. Reapplying the same procedure on the remaining nodes, we can make $K-1$ of them have a type of $(K-2)/d$. We keep on iterating till we get the type distribution $\hat{q}$ such that q equals K/d on exactly $n-K$ players, and is injective when restricted to the remaining players taking values in $\{0, 1/d, \cdots, (K-1)/d\}$.

Can we find a type distribution q that belongs to Q_n^K such that $||q||_1 < ||\hat{q}||_1$. Suppose we can, then building on the tools of Chapter 6 on complete graphs, necessarily an action configuration in $\mathcal{A}_n^K$ is a fixed-point contradicting the fact that q belongs to Q_n^K. □

To end this chapter, we give a small piece of insight. High degree nodes lower the resilience measure in the graph. One manifestation of this fact lies in the examples that meet the bounds. However, if we consider the complete graph, it has a resilience measure of 1 for $K = 1$ that grows linearly to $n/2$ for $K = n$. This said, although high degree nodes increase the resilience of a network, having a large number of high degree nodes in the network makes the network more fragile against large perturbation, and hence it is more costly to ensure its recovery.

Chapter 8

Conclusion

8.1 Summary

In this thesis, we considered a linear threshold model where agents are allowed to switch their actions multiple times. We focused on characterizing the behavior of the dynamics.

We established that in the limit, the agents in the network cycle among action profiles. We studied the lengths of such cycles, and the required number of time steps needed to reach such cycles. In particular, we showed that for *any graph structure* and *any threshold distribution* over the agents, such cycles consist of a most two action profiles. Namely, in the limit, each agent either always plays one specific action or switches action at every single time step. We also extend those results to multigraph structures over the players. We also showed that over *all graph structure* (of size n) and *all threshold distributions* no more than mn^2 time steps are required to reach such cycles, where m is some integer. We also improve convergence time results to be not more than n steps when the underlying graph is either a cycle graph, a complete graph or a tree. Our methods follow a combinatorial approach, and are based on two techniques: transforming the general graph structure into a bipartite structure, and transforming the parallel dynamics on this bipartite structure into sequential dynamics.

We also studied the problem of counting the number of cycles (fixed-points and non-degenerate cycles), the number of fixed-points and the number of non-degenerate cycles. We showed that those counting problems are #P-Complete. We further showed that deciding whether an action profile is reachable is NP-Complete and that counting the number of predecessors (preceding action configurations) of a reachable action configuration is #P-Complete.

Finally, in the setting of resilience of networks, we defined a measure μ^K that captures the minimal cost of threshold investment required to recover the network G from a perturbation of magnitude K, whereby we suppose that K agents will initially deviate from action $\mathbb{W}$ and play action $\mathbb{B}$. We show that this measure is lower-bounded by 1, and that it is upper-bounded by $n/2$, where n is the size of the network. We finally provide an interpretation of how this measure varies with respect to the network structures. High degree nodes add resilience to the network, however too many high degree nodes can make the network fragile against strong perturbations.

8.2 Future Directions

There are several questions that could be undertook, however we shall keep the list brief. We consider each chapter, and provide one or two questions that could be further developed. In Chapter 4, it would be interesting to provide a characterization of the initial configurations that lead to non-degenerate cycles and those that lead to fixed points. In Chapter 5, we believe we can improve the convergence time bound for general graphs to be linear in the size of the network. In Chapter 6, it would be interesting to characterize subclasses of $\mathcal{G}_n \times \mathcal{K}_n$ where the counting problems are tractable and devise approximation algorithms if possible for the hard cases. We need to compute resilience measures for more graph structures in Chapter 7, and mainly devise a systematic way to compute the measure. We may further consider different cost functions on the type distribution.

Bibliography

[1] M. Granovetter, "Threshold models of collective behavior", *The American Journal of Sociology*, vol. 83, no. 6, pp. 1420-1443, May 1978.

[2] S. Morris, "Contagion", *Review of Economic Studies*, no. 67, pp. 57-78, 2000.

[3] J. Kleinberg. "Cascading Behavior in Networks: Algorithmic and Economic Issues". *In Algorithmic Game Theory (N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani, eds.)*, Cambridge University Press, 2007.

[4] D. Acemoglu, A. Ozdaglar, and M. E. Yildiz, "Diffusion of Innovations in Social Networks", *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2011.

[5] D. Kempe, J. Kleinberg, E. Tardos. "Maximizing the Spread of Influence through a Social Network". *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.

[6] D. J. Watts, "A simple model of global cascades on random networks", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 9, pp. 5766-5771, April 2002.

[7] M. Lelarge, "Diffusion and Cascading Behavior in Random Networks", Feb. 2011. [Online]. Available: http://arxiv.org/abs/1012.2062

[8] L. Blume, D. Easley, J. Kleinberg, R. Kleinberg, E. Tardos. "Which Networks Are Least Susceptible to Cascading Failures?" *Proc. 52nd IEEE Symposium on Foundations of Computer Science*, 2011.

[9] H. P. Young, "The Diffusion of Innovations in Social Networks", in *Economy as an evolving complex system.* Oxford University Press US, 2006, vol. 3, pp. 267-282.

[10] A. Montanari and A. Saberi, "The spread of innovations in social networks", *Proceedings of the National Academy of Sciences*, vol. 107, no. 47, pp. 20196-20201, Nov. 2010.

[11] T. Liggett, *Interacting Particle Systems.* Springer, 1985.

[12] R. Durrett, *Lecture Notes on Particle Systems and Percolation.* Wadsworth Publishing, 1988.

[13] E. Goles-Chacc, "Comportement oscillatoire d'une famille d'automates cellulaires non uniformes", Thèse IMAG, Grenoble, 1980.

[14] S. Cook, "The Complexity of Theorem Proving Procedures", *Proc. 3rd ACM Symp. on Theory of Computing*, 151-158, 1971.

[15] L. Valiant, "The Complexity of Computing the Permanent", *Theoretical Computer Science*, vol. 8, issue 2, pp. 189-201, 1979.

[16] S. P. Vadhan, "The Complexity of Counting in Sparse, Regular, and Planar Graphs", *SIAM Journal of Computing*, vol 31, issue 2, pp. 398-427, 2001.

[17] M. Sipser, *Introduction to the Theory of Computation*, Second Edition. Course Technology, 2005.

[18] J. Munkres, *Topology*, Second Edition. Prentice Hall, 2000.