UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Bachelor Degree in Informatics Engineering

Specialization in Computing

Network Creation Games: Anarchy and Dynamics

Author:

HOMS SOLER,
Martí

Supervisor:

ÀLVAREZ FAURA,
Carme
DEPARTMENT OF COMPUTER SCIENCE

Academic Year 2017/2018
Date 24/10/2018

ABSTRACT

In this project we conduct theoretical and empirical research of topological properties for the classic model of Network Creation Games introduced by Fabrikant et al. around the Tree Conjecture and the Constat PoA Conjecture. In this game every agent buys links at a pre-established constant price, $\alpha > 0$, in order to form a connected network with the rest of the n agents while minimizing individual costs. The network induced by the players actions can be visualized as the graph $G[S] = (V, E)$, where $S = (s_1, s_2, \ldots, s_n)$ is the strategy profile of the game, $V = \{i | \forall i \in N\}$ and $E = \{(i, j) | i \in s_i \vee j \in s_j\}$. In this framework the Tree Conjecture proposed by Fabrikant et al. states that every Nash Equilibrium network for $\alpha > n$ is a tree. Also, for this setting Demaine et al. conjectured that every network has a constant Price of Anarchy. Interestingly, since Fabrikant et al. proved that the PoA of trees is less 5, we have that if the Tree Conjecture were to be true then, the Constant Price of Anarchy Conjecture would be also true for $\alpha > n$.

So far it has been proven that every Nash Equilibrium network is a tree when $\alpha > 4n-13$ by Bilò et al. The Price of Anarchy has been proven to be constant for an interval; the upper bound is when $\alpha > 4n - 13$ and the lower bound is when $\alpha = O(n^{1-\delta})$ with $\delta \geq \frac{1}{log\ n}$. In this project we study different topologies with unbounded sizes by utilizing what we call *Node Weights* and we prove that, for a certain edge direction, the Petersen graph is Nash Equilibrium when $\alpha = \frac{2n}{5}$. We also analyze other topologies under the Greedy Equilibrium defined by Pascal Lenzner and manage to find distinct equilibria for the majority of the them.

RESUM

En aquest projecte hem dut a terme recerca teòrica i empírica de propietats topològiques sobre el model clàssic de Jocs de Creació de Xarxes introduït per Fabrikant et al. al voltant de la conjectura de l'arbre i la conjectura del PoA constant. En aquest joc cada agent compra enllaços a un preestablert cost constant, $\alpha > 0$, per tal de formar una xarxa connexa a la resta dels n agents tot minimitzant costos individuals. La xarxa iduïda per les accions dels jugadors pot ser vista com el graf $G[S] = (V, E)$, on $S = (s_1, s_2, \ldots, s_n)$ és l'estratègia que configura el joc, $V = \{i | \forall i \in N\}$ i $E = \{(i,j) | i \in s_i \vee j \in s_j\}$. En aquest model la conjectura de l'arbre proposada per Fabrikant et al. afirmen que tota xarxa que està en Equilibri de Nash quan $\alpha > n$ és un arbre. També, per aquest model, Demaine et al. van conjecturar que tota xarxa té un preu de l'anarquia constant. Interessantment, com que Fabrikant et al. va demostrar que el PoA per els arbres pot ser acotat superiorment per 5, en el cas que la conjectura de l'arbre fós certa tindriem que la conjectura del PoA constant també seria certa per $\alpha > n$.

Fins ara s'ha demostrat que tot Equilibri de Nash és arbre quan $\alpha > 4n - 13$ per part de Bilò et al. El preu de l'anarquia també ha estat demostrat que és constant per l'interval acotat inferiorment quan $\alpha = O(n^{1-\delta})$ amb $\delta \geq \frac{1}{log\ n}$ i superiorment quan $\alpha > 4n - 13$. En aquest projecte estudiem diferents topologies amb tamanys no acotats utiltzant el que nosaltres anomenem *Node Weights* i demostrem que, per una certa direccionalitat, el graf de Petersen és Equilibri de Nash quan $\alpha = \frac{2n}{5}$. També analitzem altres topologies amb altres definicions d'equilibri com per exemple el Greedy Equilibri, definit per Pascal Lenzner, i aconseguim trobar equilibris per la majoria de les topologies mostrades.

RESUMEN

En este proyecto hemos llevado a cabo investigación teórica y empírica de propiedades topológicas sobre el modelo clássico de Juegos de Creación de Redes introducido por Fabricant et al. alrededor de la Conjetura del Arbol y la Conjetura del PoA Constante. En este juego cada agente compra enlaces a un preestablecido coste constante, $\alpha > 0$, para formar una red conexa con los otros n agentes mientras minimiza costes individuales. La red inducida por la acciones de los jugadores puede ser vista como el grafo $G[S] = (V, E)$, donde $S = (s_1, s_2, \ldots, s_n)$ es la estrategia que configura el juego, $V = \{i | \forall i \in N\}$ y $E = \{(i, j) | i \in s_i \vee j \in s_j\}$. En este modelo de juego la Conjetura del Arbol propuesta por Fabrikant et al. afirma que toda red que esta en Equilibrio de Nash con $\alpha > n$ es un árbol. También, por este modelo, Demaine et al. conjeturó que toda red tiene un precio de la anarquía constante. Interesantemente, como Fabrikant et al. demostró que el PoA para los árboles puede ser acotado superiormente por 5, en el caso de que la Conjetura del Arbol fuera cierta tendríamos que la Conjetura del PoA Constante también sería cierta para $\alpha > n$.

Hasta ahora se ha demostrado que todo Equilibrio de Nash es árbol cuando $\alpha > 4n - 13$ por parte de Bilò et al. El precio de la anarquia también ha sido demostrado que es constante para el intervalo acotado inferiormente cuando $\alpha = O(n^{1-\delta})$ con $\delta \geq \frac{1}{log\ n}$ y superiormente cuando $\alpha > 4n - 13$. En este proyecto estudiamos diferentes topologias con tamaños no acotados utilizando lo que nosotros llamamos *Node Weights* y demostramos que, para una cierta direccionalidad, el grafo de Petersen es Equilibrio de Nash cuando $\alpha = \frac{2n}{5}$. También analizamos otras topologias con diferentes definiciones de equilibrio como el Equilibrio Greedy, definido por Pascal Lenzner, y conseguimos encontrar equilibrios para la mayoria de topologias mostradas.

Contents

1 Introduction

In the following chapters of this document I will explain the object of this project, the motivation in conducting research, the objectives and everything else needed to get a clear idea of what this manuscript, and essentially what this project, is about.

1.1 Objectives, Motivation and Benefits

The objective of this project is to close my Degree and open a way to the research world by conducting theoretical and experimental research on a field close to the Informatics Engineering Degree, Algorithmic Game Theory. We will define in detail our research once we have introduced the necessary preliminary definitions.

As for the motivation is concerned, my interest in learning new things is what brought me to choose scientific research as my Degree's Final Project. This kind of project, though it has an experimental part, is based really hard on theoretical study and we cannot speculate about the possible outcomes of such research, since, as the core of all research is, we don't know what we can achieve.

Nonetheless, we are certain that there will be benefit to be extracted from this project, from a personal to a more general point of view. We have the hope that the results can help steer the direction of future work and, in a more subtle way, push the research ahead. The real idea behind all this commitment is to improve the understanding of what we don't know.

1.2 Object of the project

The object of study of this project is the Tree Conjecture and the Constant PoA Conjecture around the Sum Classic Model of Network Creation Games. This game serves as a mathematical model of network creation between selfish agents.

To begin with, the Tree Conjecture states the common sense that in a game where links have a price, the higher the price of the links the fewer connections the agents will have between them.

Finally, the Constant PoA Conjecture states that the ratio between the social cost of an equilibrium configuration and the social optimum is constant. Meaning that by selfishly playing the game all the equilibria configurations are close to the optimum configuration.

1.3 Project outline

This document structure is the following one:

- In Chapter 2 we present a summarized glossary of the definitions about Graph Theory that we use the most.
- In Chapter 3 we introduce the necessary definitions and concepts related with Algorithmic Game Theory.
- In Chapter 4 we present an empirical study based in the use of computer algorithms to generate and analyze topologies.
- In Chapter 5 we dive deeper into small diameter topologies and change the computer for pen and paper.
- In Chapter 6 we study the topologies under different definitions of equilibria in order to gather more information on their peculiarities.
- Finally, in Chapter 7 we present an overall conclusion of all the experiments, findings, properties and ideas that we have gathered through-out the different experiments and we also conjecture about hypothetical future work lines.
- In Appendix A we explain the planning of the project, its sustainability, costs, social and environmental risks and impact.

2 Preliminary Definitions

Since this project uses Graph Theory to represent the game models we present here a summarized glossary of all the most important definitions that appear in the following chapters. Ad hoc definitions for concepts that are not recurrent in the experiments will be given in situ where the said experiment is explained.

Definition 1. ***Cycle:*** *Let* $G = (V, E)$ *a graph.* G *has a cycle if* $\exists u, v \in V$ *such as there are at least two different paths going from one vertex to another.*

Definition 2. ***Tree:*** *A graph is called a tree-graph is it has no cycles.*

Definition 3. ***Star (l):*** *A graph is called l-star if it is a tree with exactly one central node and l external nodes with an edge towards the central node.*

Definition 4. ***Clique (k):*** *A k-clique* C *is a subset of vertices* $C \subseteq V$ *of a graph* $G = (V, E)$ *such that* $|C| = k$ *and* $\forall v, u \in C$*, if* $v \neq u$ *then,* $(u, v) \in E$*.*

Definition 5. ***2-vertex-connected:*** *We say that a graph is 2-vertex-connected if and only if for each node v, when removing v and all the edges incident with v from the original graph, the new graph remains connected. Note that from now on, we use 2-vertex-connected and biconnected indistinguishably.*

Definition 6. ***Girth:*** *The girth of a graph* G *is the length of the shortest cycle found inside* G*. In the case that* G *doesn't have any cycles in it, the girth is said to be infinity. As an example an hexagon has girth equal to 6.*

Definition 7. ***Maximal:*** *A subgraph* $G' \subseteq G$ *is maximal for a certain property if it has such property and no other graph* G''*, such as* G' *is a subgraph of* G''*, satisfies the same property.*

3 Network Creation Games

In this chapter we provide the basic definitions and main concepts needed to understand this project and its contribution. Also, in order to frame the importance and hardness of the problems studied in this work, we present the related work done around them.

3.1 Strategic Games

Strategic Games is the general kind of games inside *Game Theory* that our project works on. *Network Creation Games* are a sub-type of *Strategic Game* and, therefore, by defining the general concept and introducing some examples we intend to ease this chapter's comprehension.

Definition 8. *A Strategic Game is defined by three main components; the players which we define as rational and selfish agents, their strategies which can be seen as a set of determinate actions they choose to perform and a cost function that represents the cost for each player and his/her strategy. Therefore, a Strategic Game is formally defined as a tuple* $\Gamma = (N, (S_i)_{i \in N}, (c_i)_{i \in N})$*, where* $N = \{1, 2, \ldots, n\}$ *is the set of players. Each player* $i \in N$ *has a set of available actions or strategies* S_i*, and in any configuration of a game, every player has to choose an available strategy* $s_i \in S_i$*. It is important to point out that each player selects the strategy without the knowledge of the other players selections. Hence, the result of a game is the collection of the strategies selected by the players, known as the strategy profile of a game.*

When we talk about a strategy profile we are referring to a specific configuration of the element $s = (s_1, s_2, \ldots, s_n)$*. We use the notation* $S = S_1 \times \cdots \times S_n$ *to denote all the possible strategy profiles for a game and* $s' = (s_{-i}, s'_i)$ *to represent that a strategy* s' *differs from* s *only because of the player* $i \in N$*, where* $s' = (s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_n)$*.*

The cost function $c_i(s) : S \to \mathbb{Q}$ *represents the amount that has to be paid by player* $i \in N$ *when the configuration of a game is defined by the strategy profile* s*. Consequently, the cost of player* $i \in N$ *does not depend only on his/her strategy but rather on the decisions of all the players.*

An example. One of the most famous classical examples in Game Theory is the strategic game called "Prisoner's Dilemma". This game is used to clearly represent that sometimes rational agents do not cooperate even-though it appears to be in their best interest. This game is presented as follows:

"Two criminals, we will call them A and B, are arrested and subject to interrogation separately with no means of communicating with one another. The prosecutor doesn't have enough evidence to convict either of them for the hardest penalty, but she has enough for the lesser penalty. Then she offers the possibility to reach a deal with both prisoners. Each prisoner is given the chance to testify against his/her colleague and lesser his/her penalty or remain silent."

The outcome of the offer looks like this:

- If A and B both testify against one another, both of them serve 5 years in prison.
- If one of them testifies and the other remains silent, the prisoner that testified will serve 1 year in prison weather the other will serve 7 years.
- If A and B both remain silent, they will only serve 2 year in prison .

A $\backslash B$	B stays silent	B testifies
A remains silent	2 \2	7 \1
A testifies	1 \7	5 \5

Table 1: This table shows the outcomes for the players in each strategy profile

We have that both of them have the same available actions which are "T" for testifying and "S" for staying silent, and therefore we can define $S_i = [T, S]$, where $i \in \{1, 2\}$. We can use the Table 1 as a reference and calculate the outcomes for both prisoners. To begin with, if only one of them testifies then, the snitch will serve one year and the betrayed will serve seven, therefore $c_1(T, S) = c_2(S, T) = 1$ and $c_1(S, T) = c_2(T, S) = 7$. If both of them testify against one another they serve five years and the outcomes are $c_1(T, T) = c_2(T, T) = 5$. To conclude, if both of them remain silent then, the penalty will be two years in prison and the outcomes are $c_1(S, S) = c_2(S, S) = 2$.

From a rational and selfish point of view it is clear that testifying against the partner is far better than to remain silent. On the one hand, if one thinks that the partner will remain silent then, by testifying, one can reduce the penalty in one year because the penalty is only 1 year instead of 2. On the other hand, if one thinks that the partner will testify then, once again, testifying shortens the penalty because now it goes from 7 to 5 years.

3.1.1 Nash Equilibrium

An important concept related with *Game Theory* is the *Nash Equilibrium.* In some games there is a state that resembles an *equilibrium*, an idea of balance between players as a state or structure where no player has incentive to change his/her strategy because by doing so he/she does not improve his/her benefit. This concept is what is known as *Nash Equilibrium*, named after the famous American mathematician John Forbes Nash Jr. For sake of simplicity we will refer a *Nash Equilibrium* with NE or just *equilibrium.* Formally, NE is defined as follows:

Definition 9. *Let* $\Gamma = (N, (S_i)_{i\in N}, (c_i)_{i\in N})$ *be a strategic game. The strategy profile* $s \in S$ *is a NE iff for all* $i \in N$ *and for all deviations* $s' = (s_{-i}, s'_i)$ *relative to the player* $i \in N$*, we have that* $c_i(s') \geq c_i(s)$*, or what it is the same* $\Delta C_i(s') = c_i(s') - c_i(s) \geq 0$*. Meaning that all agents do not have an interest on changing his/her strategy individually because the cost difference of the* *new strategy over the old one is greater than zero, implying that changing strategies will result in a more expensive state than the one they had before.*

An example. In the "Prisoner's Dilemma" game we know that the configuration $s = [T, T]$ is a NE, because no player has incentive to change unilaterally his/her strategy. If one of them changes his/her strategy the penalty will be modified from serving 5 years to serving 7 years, which is not a desirable outcome. From the four possible game configurations $[S, S], [S, T], [T, S], [T, T]$ the latter one is the only *equilibrium*.

3.1.2 Price of Anarchy

Since the game we are working on considers individual and selfish choices it is interesting to introduce the concepts of *Social Cost* and *Price of Anarchy.*

Definition 10. *Let* $\Gamma = (N, (S_i)_{i \in N}, (c_i)_{i \in N})$ *be a strategic game. Given a strategy profile* $s \in S$ *we have that the social cost* $c(s) = \sum_{i \in N} c_i(s)$. *Therefore, the social cost of a strategy profile is the sum of all individual costs.*

An example. Once again in the "Prisoner's Dilemma" game we can see a desirable outcome for both players, what its called *Best Social Cost.* We know that if they could communicate with one another the would decide to remain silent and therefore serve only two years, $s = [S, S]$. However we know that the natural *equilibrium* reached in this particular game is $s = [T, T]$. If we compare the social cost of all configurations, $c(S, S) = 4$, $c(T, S) = c(S, T) = 8$ and $c(T, T) = 10$, we can clearly see that the first one is a more socially desired configuration since all the players would be far better than in the other ones.

The *Price of Anarchy*, or just *PoA* for simplicity, is a form to measure how the social efficiency of a system diminishes due to the selfishness of the players behaviour. Since a NE is reached by following the selfish choices of the players, the *Social Cost* of such configuration may not be an optimal one, instead a *Social Cost* is usually reached with a controlled configuration, meaning that the players strategies are chosen by a centralized agent. The *PoA* is defined as the ration between the worst *equilibrium* and the best *Social Cost.*

Definition 11. *Let* $\Gamma = (N, (S_i)_{i \in N}, (c_i)_{i \in N})$ *be a strategic game and let* $E \subseteq S$ *be the set of Nash Equilibria. Then the Price of anarchy is* $PoA = \frac{max_{s \in E} c(s)}{min_{s \in S} c(s)}$.

An example. Finally we can finish the game analysis by computing the PoA. Since in this game there is only one equilibrium, [T,T], we already know that its social cost is 10. We also know the configuration with the best social cost, [S,S], which is 4. Therefore the $PoA = \frac{10}{4} = 2.5$, meaning that the worst equilibrium found is more than two times worse than the best possible social configuration.

3.1.3 Dynamics

It is obvious that this type of games do not define the entire spectrum of models studied by *Game Theory*. This type of games, where the player strategy selection is done at the same time is perfect to define some games like rock-paper-scissors but it is not adequate to define continuous games. To study this types of games we consider the *Dynamics of a Game*.

Definition 12. ***Dynamics of a Game****: The Dynamics of a Game can be seen as a graph where each node s represents a strategy profile defining a certain configuration of the game and (s, s') is an edge of the dynamics graph if $s = (s_{-i}, s_i)$ and $s' = (s_{-i}, s'_i)$ for some player i and $s_i \neq s'_i$.*

In this project we use an algorithm to *Walk the Dynamics Graph*, and it works as follows. Starting from an initial configuration of a game with strategy profile s, we can define a node in the dynamics graph that represents this configuration and call it G_0. At this point we can perform a step from G_0 to G_1 that consists of changing the strategy of exactly one player, any player. By doing so we can decide a turn for the players to improve his/her current strategy and progress from the initial configuration network G_0 to the network G_l in l turns. As seen in Figure 1 and as explained above, the move of the player i in G_{l+1} is the replacement of the strategy of that agent, s_i, for another suitable strategy, s'_i. The new strategy profile of the game in G_{l+1} is $s' = (s_{-i}, s'_i)$. The cost of a player in G_l will depend on the configuration of the game in that state. We can continue with this dynamic process until we reach an Equilibrium configuration or what its know as a sink in the dynamics graph, a cul-de-sac.

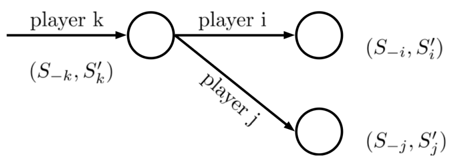


Figure 1: Example of a transition between different states in the Dynamics of a game

We can introduce the concept of Best Response as the policy to compute the best strategy that the players can choose from all the other ones.

Definition 13. *Let $\Gamma = (N, (S_i)_{i\in N}, (c_i)_{i\in N})$ be a strategic game. We define the set of best strategies of player i given a strategy profile $s \in S$ as: $BR(i, s) = [arg\, min_{s_i \in S_i} c_i(s_i, s_{-i})]$. Therefore the Best Response can be seen as the strategy, or set of strategies, which results in the best outcome for a player, the best payoff, and minimizes the cost function of that player.*

3.2 Sum-Model in Network Creation Games

Network Creation Games is a *strategic game* that studies networks, like Internet, with selfish agents and no central regulation. In this model we study how networks are created and how they evolve, they structure and their properties. Since this project studies the *Sum Classic* model we will define it and present the related work done so far on this topic.

Sum Classic Model: [6] The *Sum Classic* model is a strategic game that can be defined by a rational parameter $\alpha \in \mathbb{Q}^+$ and a set of players N. In this model each player indicates the subset of nodes that he/she wants to be connected to, this subset are the links bought by the player. The network associated to a strategy profile s is defined by the graph $G[S] = (V, E)$, where $V = \{i | \forall i \in N\}$ and $E = \{(i,j) | i \in s_i \vee j \in s_j\}$. This network can be seen as directed and undirected at the same time. We want to consider the directed version when we are focused on the strategies of the players that define the overall communication network. And we focus on the undirected version when we desire to analyze the topology of the network and its properties.

The cost associated to a strategy profile s for a player $u \in V$ is defined as follows:

$$c_u(s) = \begin{cases} \infty, & \textit{if } G[S] \textit{ is not connected} \\ \alpha|s_u| + \sum_{v \neq u} d_{G[S]}(u,v), & \text{otherwise} \end{cases}$$

Where $d_{G[S]}(u,v)$ is the distance between u and v in either the undirected graph $G[S]$. Notice that the cost contains the bought links and the corresponding quality of the network associated to the strategy profile as the sum of distances between the players.

Related Work

Since Fabrikant et al, [6], first introduce this network creation game model several research lines have been explored. Nevertheless, there are a few open theoretical questions that remain to be answered. This project focuses on two conjectures, The Tree Conjecture and the Constant PoA Conjecture, and there is a connection between one another.

Begining with the original Tree Conjecture we have that Fabrikant et al, [6], states that it exists an $A > 1$ such as all the equilibrium networks for $\alpha > A$ are trees. In a game where links have a price, it is common sense to think that the bigger the price the less amount of links there would be in a network; the Tree Conjecture states this common sense by basically saying that NE networks when the cost per link is high are trees. In [6] the authors prove that the PoA when $\alpha > n^2$ is $O(\sqrt{\alpha})$ and that the PoA of a NE tree can be upper-bounded by 5, meaning that if the Tree Conjecture were to be true, the Constant PoA Conjecture would be true too.

Not long after Albers et al. in [1] disprove the orignal Tree Conjecture when $\alpha < n$ by giving a counter-example. Furthermore, the authors prove that the PoA is $\Theta(1)$ for $\alpha \geq 12n(logn)$. After it, Demaine et al. in [5] improves Albers results by upper-bounding the PoA to $2^{O(\sqrt{logn})}$ when $\alpha < 12n(logn)$ and proving that the PoA is constant when $\alpha = O(n^{1-\delta})$ with $\delta \geq \frac{1}{log\ n}$.

$\alpha =$	0	1	2	$\sqrt[3]{n/2}$	$\sqrt{n/2}$	$O(n^{1-\epsilon})$	$4n-13$	$9n$	$17n$	$65n$	$12nlog\ n$	∞
PoA	1	$\leq \frac{4}{3}$ [6]	≤ 4 [5]	≤ 6 [5]	$\Theta(1)$ [5]	$2^{O(\sqrt{log\ n})}$ [5]	$\Theta(1)$	$\Theta(1)$	< 5	< 5 [4]	1.5 [1]	

Table 2: Graphic representation of the PoA intervals found so far.

Then, Mihalák & Schlegel in [13] focus on $\alpha > n$ and prove that for $\alpha > 273n$ all NE networks are trees. Five years later Mamageishvili et al. in [10] enhance the known intervals when $\alpha \geq 65n$. Recently, not even two years ago, Àlvarez & Messegué in [2] lower the boundary for the Tree Conjecture for $\alpha > 17n$ showing that all NE are trees and they also prove that for $\alpha > 9n$ all NE had $PoA = O(1)$.

Very recently Davide Bilò and Pascal Lenzner in [4] show that all NE are trees when $\alpha > 4n - 13$ improving Àlvarez & Messegué's work. And finally, this year, Àlvarez & Messegué, [3], improved even further the previous results by proving that for NE networks when $\alpha > n(1 + \epsilon)$ the PoA is constant, when ϵ is any small positive constant. They also prove that for $\alpha > n$ all NE networks are trees when their diameter is higher than a prefixed constant.

As can be seen these problems are heated. The principal questions have been improved by a lot in the last few years but they remain open for a certain interval, $n < \alpha < 4n - 13$ for the Tree Conjecture and $\alpha = O(n^{1-\delta})$ for the lower bound and $\alpha < n(1 + \epsilon)$ for the upper bound on the Constant PoA Conjecture.

3.3 What happens when α is near to n?

The aim of this project is to study the behaviour of graph topologies when the α value is near to n. Is the *Tree Conjecture* true for all $\alpha \geq n$? Is the Constant PoA Conjecture true? In order to try and clarify these questions we conduct several experiments inside our interval of interest, researching different network topologies and also considering different soft notions of equilibria. This way the project can be divided in two distinguishable parts.

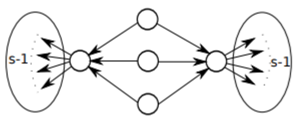


Figure 2: Non-tree NE when $\alpha = n - 3$ and $n = 2s + 3$
[10]

In the first part of this project we examine different topologies and search for properties of NE networks. Our interest is in the right part, when $\alpha \geq n$, but we also want to dip as much as we can on the left part, when $\alpha < n$ but α close to n. For this section we use the knowledge of several papers and manuscripts alongside fewer examples of NE graph that aren't trees when $\alpha = \Theta(n)$; those examples can bee seen in Figure 2 and Figure 3. To conclude we will have several topologies which we want to inspect under different restrictions.

In the second part of this project we focus on different definitions of *equilibria*. One of the new definitions of *equilibria* is the so called *Greedy Equilibrium*, defined by Pascal Lenzner in [7] and in [9]. We know that if G graph is NE then, G is GE and the other soft-equilibria. This implication doesn't work in the other direction but, on the one hand, if we find that G, a non-tree graph, is in equilibrium for a different definition of equilibria maybe it can help us find a G, non-tree graph, which is in NE. On the other hand, if we cannot find non-tree equilibria then, it reinforces the Tree Conjecture. Our goal is to find an equilibrium for all the topologies designed so far.

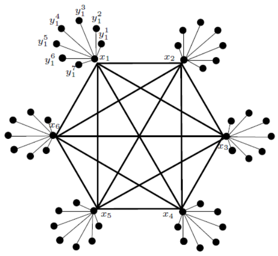


Figure 3: Non-tree NE when G is a (k, l) clique of stars, all the edges are bought by the clique vertices, no edge is bought twice and $\alpha = \frac{n}{k}$.

[1]

4 Empirical study of NE topologies when $\alpha = n$

In this chapter we conduct experimental research to explore the topologies of the NE networks when the α value is equal to n. We center our efforts on the Tree Conjecture and the Constant PoA Conjecture and we are interested in non-tree NE topologies. Even-though in this chapter we only center ourselves when $\alpha = n$, we still use restrictions on topologies when $\alpha > n$ because the results of this section will be used in the following experiments and analysis.

We focus on topologies with low edge density and small diameter, what we call almost-tree-like topologies, due to theoretical results which state that $\forall \epsilon \exists c$ such as $\alpha > n(1+\epsilon)$ and $diam(G) > c$ if G NE then G is a tree [3].

4.1 First Approach

The first setting of experiments consists of the generation of an initial network, its posterior evaluation of whether it's a NE or not and a analyze the problems we face to help us in the following experiments. With this first approach we want to test different topologies without imposing any restrictions. To generate the initial networks we have two different algorithms, exhaustive and random. For the evaluation of the network we use Algorithm 1. This algorithm consists in sequentially checking if players are applying their best-response and if one of them can improve his/her current strategy then, the network is not NE.

Exhaustive Exploration

For the exhaustive exploration we use a backtracking algorithm to generate the full space of networks of size n. In order for the backtracking to work properly it's very important to detect all the different cases:

$$\forall (u,v) \in E \begin{cases} (\text{Case 1: })(u,v) \notin E \\ (\text{Case 2: })(u,v) \in E \text{ and } (u,v)_{owner} = u \\ (\text{Case 3: })(u,v) \in E \text{ and } (u,v)_{owner} = v \end{cases}$$

In this generations we set the number of vertices and generate exhaustively all the graphs with that size. Therefore we have to contemplate that an edge can belong to the final graph in two different directions or not belong. We set the α value to be equal to n, considering $n \in \{3,4,5\}$. Here, in particular, we present the pseudo-code of the algorithm.

Conclusions: This experiment is a small step towards consolidating our intuition on the problems. Now we can say that there are no NE networks when $\alpha = n$ and $n \in \{3,4,5\}$. It may seem a small contribution but now this result can be improved with future work and no-one needs to "re-invent the wheel".

Algorithm 1 Network evaluation

```
function NETWORK_EVALUATION(G)                                              ▷
    n = G.nodes().size()
    return is_nash_equilibrium(G, n)
end function

function IS_NASH_EQUILIBRIUM(G, alpha)
    for i in G.nodes() do
        cost_before = calculate_cost(G, i, alpha)
        if is_exerting_best_response(G, i, cost_before, alpha) then
            return False
        end if
    end for
    return True
end function

function IS_EXERTING_BEST_RESPONSE(G, player, cost_before, alpha)
    Strategies = all_strategies(G, player, cost_before, alpha)    ▷ returns all the strategies
possible for the players the set [Δc, Strategy]
    Strategies = sorted(Strategies)                      ▷ Sorts the set increasingly by Δc
    if Strategies[0][0] < 0 then
        return False    ▷ Since the strategies are sorted from lower higher cost difference if the
first one has a negative cost difference we know the player is not exerting his/her best response
    end if
    return True
end function
```

Algorithm 2 Exhaustive Generation

```
function EXHAUSTIVE_GEN(n)                          ▷ Where n is the size of the network
    G = empty_graph(n)
    E = K_graph(n).edges()                          ▷ List of the edges in the K-graph of size n
    return exhaustive_backtracking(G, E)
end function

function EXHAUSTIVE_BACKTRACKING(G, E)
    if E is empty then
        return network_evaluation(G)
    else
        edge = E.pop()
        exhaustive_backtracking(G, E)                                   ▷ Case (1)
        G.add_edge(G, edge[0], edge[1], owner = edge[0])
        exhaustive_backtracking(G, E)                                   ▷ Case (2)
        G.add_edge(G, edge[0], edge[1], owner = edge[1])
        exhaustive_backtracking(G, E)                                   ▷ Case (3)
    end if
end function
```

It also helps us to figure out how other experiments can be. With this brute-force generation we have experienced computational limitations when increasing the size of the network. We need to find a system to explore multiple topologies with a single generation.

Random Exploration

For the random exploration we use a random algorithm to generate graphs with uniform probability. Note that if an edge appears to be owned by two nodes, the topology isn't NE and we can discard it directly before evaluating the network, (1). Here we present the pseudo-code of the algorithm.

For the random generation we consider network sizes ranging from 10 to 15 and, as specified in this chapter, the α parameter is set to be equal to n. We execute this random experiment in batches. Each batch generates 50 random topologies 30 times, for a total of 1,500 topologies. We run one batch for each size, $n \in \{10, 11, 12, 13, 14, 15\}$. Note that because of how the random algorithm is programmed we can discard those topologies which have an edge owned by both ends.

Conclusions: With this experiment we find that the number of batches has to be increased if we want to generate enough topologies to analyze. We decide not to increase the dimensions of the network because, the bigger the network size is the less topologies are generated. Nontheless with the topologies generated we can state that there were no non-tree NE networks. From all the topologies generated a big part of them are not NE because of the existence of an edge owned by two different players.

Algorithm 3 Random generation

```
function RANDOM_GEN(n)                                          ▷
    G = empty_graph(n)
    for i = 0 ; i < n ; i + + do
        for j = 0 ; j ≠ i and j < n ; j + + do
            p = rand_generate(0, 1)
            if p == 1 then
                if G.exists_edge(i, j) then
                    return False                                ▷ (1)
                end if
                G.add_edge(i, j, i)              ▷ Adds edge i, j with owner i
            end if
        end for
    end for
    return network_evaluation(G)
end function
```

Results

Lastly, for this first run of experiments we can say that there are no NE networks when $\alpha = n$ and $n \in \{3, 4, 5\}$. We also have experienced some computational problems due to the amount of combinations. Our goal is to explore as many topologies as possible with the minimum amount of generations, therefore we are not going to use exhaustive generation from now on. For the next experiment we increase the number of batches for the random generation and only evaluate the topologies that have certain properties. This properties are exposed in the next part.

4.2 Restrictive Exploration

In the last experiment we experienced some computational problems due to the huge search space our question resides in. One of the solutions we choose is to impose certain properties to the topologies analyzed. We start to restrict the analyzed topologies by imposing two properties: the networks have to be 2-vertex-connected and cannot have a triangular cycle. Here we give the **definition**, reasons and proof of why we choose this restrictions.

Since we are interested in non-tree NE networks, we impose the 2-vertex-connectivity in order to impose that the analyzed networks are non-trees and have no leaves.

Lemma 1. *[11] For G, a NE network, and $\alpha = n$, there do not exist any triangle cycles in G.*

Proof. It is just a consequence of the results from [10], stating that the girth of any non-tree NE network G has length at least $2\frac{\alpha}{n} + 2$. Therefore for G, a NE network, and $\alpha = n$ the girth of G has to be $girth(G) \geq 4$ □

In this experiment we use the *Best Response Dynamics* as a tool to evaluate multiple topologies with a single generation. Note that we have defined the *Dynamics of a Game* in Chapter 3. In our experiment we define an algorithm to decide the players transitions to create a path that *Walks the Dynamics Graph.* The algorithm used to walk the dynamics implements a turn and response policy which we choose to be a sequential best-response algorithm.

- Turn policy: The players are chosen in a defined order, one after the other. A Round-Robin policy.
- Response policy: Each player computes his/her *Best Response.*

Here we present the pseudo-code of the algorithm:

Algorithm 4 Dynamics Sequential Best-Response Walk

```
function SEQ_BR_WALK(G)                                   ▷
    isNE = false
    while not isNE do
        isNE = true
        for i = 0; i < n; i++ do
            b = compute_best_response(G, i)    ▷ Returns true if the player changes his/her
strategy, false otherwise
            if b then
                isNE = false
            end if
        end for
    end while
end function
```

Our objective is to analyze the sink topologies of the dynamics. As stated in [8], we know that networks are not guaranteed to converged, ergo they can loop. In order to prevent it, we set a maximum number of steps for the network to converge in order to avoid those configurations who are too slow to do so or they simply don't do it. In every turn if a player has an incentive to improve his/her current configuration we transition to another state of the dynamics, otherwise the turn is skipped without performing any changes to the topology. A round concludes when all the players have finished their turn.

This second setting of experiments works similar as the first one. We begin by generating a network using the random algorithm, then we discard the networks that do not comply with our restrictions and finally we walk the *Dynamics Graph* with Algorithm 4.

The results presented are shown with a number inside each player, which are their "names", and a tag *owner* followed by a number for the edges, which state the player that has bought that link.

Random Exploration

For this experiment we use the Algorithm 3 explained before. We keep the network size in the same range as before, $n \in [10, 15]$ and the α parameter, setting it equal to n. This time we run 5 batches of 1,500 topologies per networks size. Note that by probability there are a lot of networks which do not have the properties we specified and even-though we have a large number of networks generated we do not have a huge amount of desired topologies, as seen in Figure 5.

Since the size of the initial configuration of a network this big made the visualization of the randomly generated graph very difficult and messy, we only include the sink topology, as seen in Figure 4.

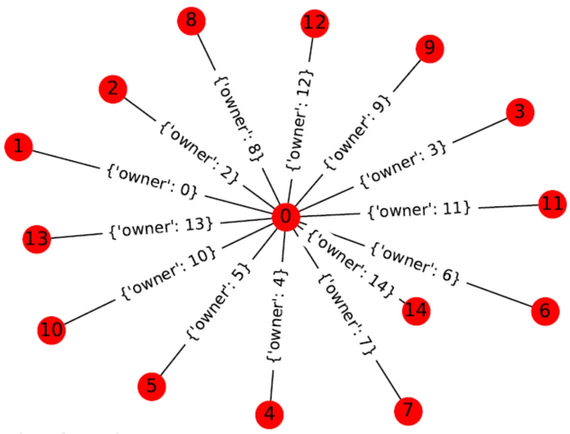


Figure 4: Sink topology of the random graph generation

The topologies analyzed have a fast convergence, as seen in Figure 5. In the graphic can be seen that because the amount of possible networks of size 10 is much smaller than the amount of networks of size 15 the topologies generated that have the properties that we impose vary. What does not vary is the overall speed of convergence of the topologies, which remains at 2 or 3 rounds.

As Pascal Lenzner states in [9]: "First there is a phase with mostly deletions. Then this phase is followed by a phase where mostly swap and some buy and deletion operations occur. This is followed by a final phase where some swaps and mostly deletions happen."

He was talking about the results obtained by running an experiment with a *Greedy Dynamics* and as it seems the number of phases looks very similar to the one that we have, and in this experiment we use *Best Response Dynamics*.

Results

To conclude this second setting of experiments we do believe that we have to work with bigger topologies. The idea of using the *Best Response Dynamics* is useful but even-though we impose the constraints specified at the beginning of this section we believe that generating the topologies

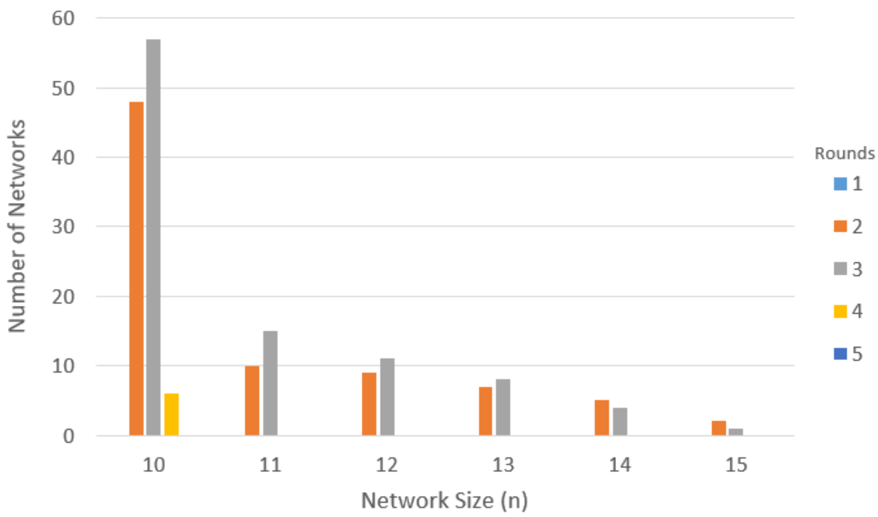


Figure 5: Graphic with the amount of rounds needed for the networks to converge. (left-axis) Amount of networks generated, (bottom-axis) size of the networks from 10 to 15 and (right-side) shows the color of the columns specifying at which round the topology reach a sink topology.

via computer algorithms is not the way to go from now on. With the results we have gathered some interesting properties and we believe that they can be used to design topologies on paper. From now on we use computer algorithms to evaluate the networks but not to generate them.

5 Topologies of small diameter

In this chapter we design topologies with certain properties that we have gathered from research and analyzing the results of previous experiments. Our goal is to find a family of NE networks that complies with all the constraints we impose.

To begin with, we restrict the type of topologies we analyze by using *Node Weights*. This restriction was inspired by the Mamageishvili, Mihalák and Müller graph, [13], that can be seen in Figure 2, and by Susanne Albers et. al graph, [1], that can be seen in Figure 3 and by Àlvarez & Messegué in [2]. In these graphs there are nodes that have bought links towards a cluster of nodes. We define "Node Weights" for 2-vertex-connected components as follows:

Definition 14. ***Node Weights:*** *In a connected graph $G = (V, E)$ we say that $H \subseteq G$ is a biconnected component of G if H is a maximal biconnected subgraph of G. For every $u \in V(H)$ we define $T(u)$ as the connected component containing u and the subgraph induced by the vertices $(V(G) \backslash V(H)) \bigcup \{u\}$. The weight of a node $u \in V(H)$ is then defined as $|T(u)|$.*

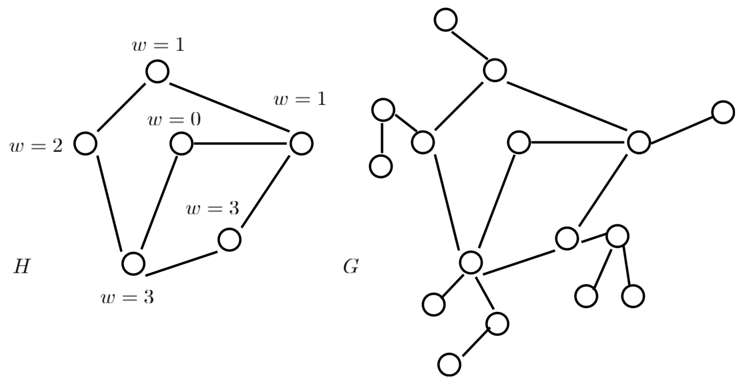


Figure 6: Example of the weight visualization of H and the general visualization of G.

To help us visualize topologies and make the designing easier we use a simple method; we take a node as our base node and divide the graph in several layers. Each layer also has a number representing the distance from it to the base node. Then, the rest of the nodes from the graph are inserted in their corresponding layer depending on the distance from that nodes to the base node. This method helped us with calculating deviations both for the H component and the G graph. A visual example of a layer-designed hexagon can be seen in Figure 7.

For this analysis we limit the size of the node weights and require that $\forall u \in V$, $|T(u)| = W$, W a certain positive value. We also impose that the subgraph hanging from each node in H is a star with the center node being the one that belongs to H and all the edges of the start are bought by the central node. For the α parameter in this particular setting of experiments we decide to not restrict it in any way. As a result of doing this we expect to find some results either in the left side when $\alpha < n$ or in the right side when $\alpha \geq n$.

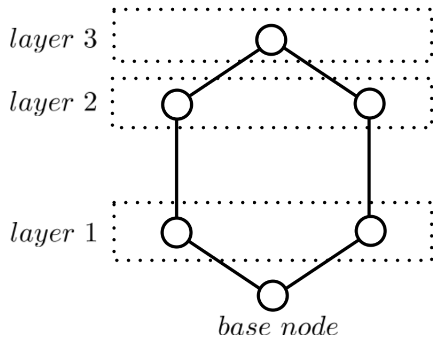


Figure 7: Example of the layer visualization of a hexagon.

We believe that the use of *Node Weights* is not enough. Because of that we present more constraints for G, if G is NE for $\alpha > n$. Detailed proofs for this constraints can be found in [11] and [12].

We are interested in small diameter and high girth topologies. Hence, we divide the properties depending on the diameter of H, the biconnected component, being it 2 or 3. Our reference topology when designing is the Petersen Graph seen in Figure 8.

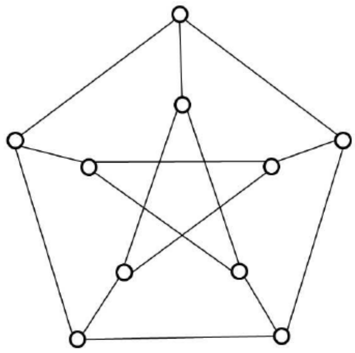


Figure 8: Original Petersen graph

Diameter 2

Let us suppose H is the maximal 2-vertex-connected subgraph of a NE graph G and $diam(H) = 2$. Let us review certain properties of H when $\alpha > n$.

Lemma 2. *[11]. For $\alpha > n$, $deg_H^+(u) \leq 2$.*

Lemma 3. *[11]. For $\alpha > n$, if $u, v \in V(H)$ are at distance two, then $deg_H^+(u) \leq 1$ or $deg_H^+(v) \leq 1$.*

Proposition 1. *[11]. There do not exist any non-tree equilibrium for $\alpha > n$ having a biconnected component of diameter exactly two.*

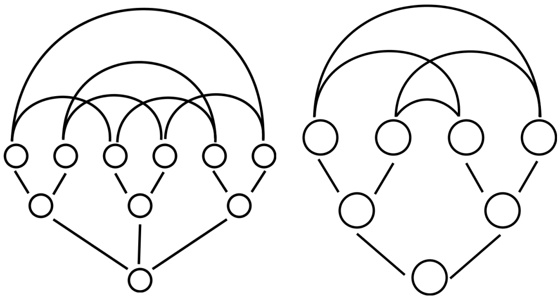


Figure 9: Manually designed topologies of diameter 2. Petersen graph (left) Petit-Cinc (right)

Topologies: For this range we have very strict properties. Because of that we have designed only two topologies, one that turned out to be the Petersen graph and another that we call Petit-Cinc, seen in Figure 9.

Notice that, since Proposition 1 restricts the existence of a NE network of diameter equal to two when $\alpha > n$, then we analyze topologies of diameter 2 when the cost per edge is smaller to the size of the network, $\alpha < n$.

Diameter 3

Now, let us suppose that H is the maximal 2-vertex-connected subgraph of G and $diam(H) = 3$. Here we present some properties of H when $\alpha > n$.

Lemma 4. *[11]. If u, v are nodes with $d(u, v) \geq 2$ such that $D_G(u) \leq D_G(v)$ and $diam_G(u) = 3$, then $deg^+(v) \leq 1$.*

Lemma 5. *[11]. There do not exist nodes $v \in V(H)$ with $deg^-(v) > 0$ and $deg(v) = 2$.*

Proposition 2. *[12]. Let H be a non-trivial biconnected component of diameter $d \geq 3$ and $girth(H) = 2d + 1$. Then, if G is a NE for $\alpha > n$, H is a cycle.*

Topologies: In this case, Proposition 2 restricts the girth of the networks to be less than the maximum. With diameter 3 the girth for the topologies that we focus on has to be 5 or 6. We know that squares and triangles are restricted because, as proved in [10], if G is a non-tree NE, when $\alpha > n$, then $girth(G) \geq 2\frac{\alpha}{n} + 2 > 4$. Considering that we do not design cycles, Proposition 2 restricts the girth of G to be $girth(G) < 7$.

For those topologies with diameter 3 we use the same starting point, the Petersen graph, but now we insert a new empty layer. Then, we populate the empty layer with nodes and connect those nodes with the second layer to ensure that the final structure has diameter 3. This new layer brings more flexibility in the designed topologies. Here, in Figure 10, you can see the results.

The four new topologies all have names to be easily identified. We will explain more in depth how we designed them in Chapter 6, when we conduct an analysis under different definitions of equilibria for all of them.

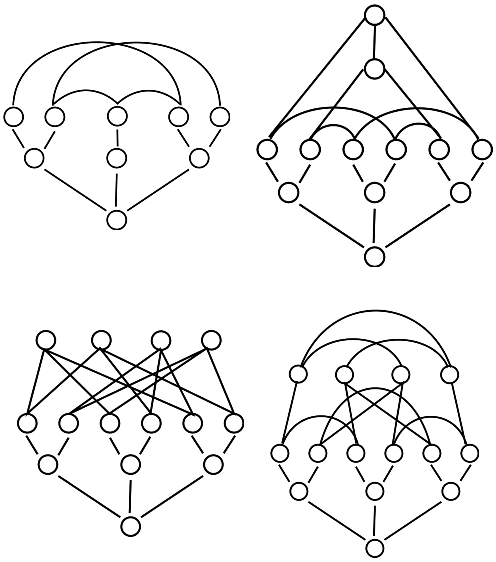


Figure 10: Manually designed topologies of diameter 3

Results

To conclude this section we have analyzed topologies under NE properties. We have found a NE topology on the left side of the interval, when $\alpha < n$. As far as the right side is concerned we haven not found any NE topologies, but we believe that a posterior study with different definitions of *equilibria* may bring some information. The Petersen Graph is the topology that we know an α value in which it is NE and here we present the same proof that can be found in [11]. I would like to remark that all the topological design and the research of properties has been done jointly with A. Messegué but the proof of Proposition 3 that we show has been written solely by A. Messegué.

Proposition 3. *[11] Let G be a graph of n nodes with $H \subseteq G$ a non-trivial bi-connected component of 10 nodes v_1, v_2, v_3, v_4, v_5 and w_1, w_2, w_3, w_4, w_5 such that $|T(v_i)| = |T(w_i)| = n/10$. The topology of each $T(x)$ is a directed star rooted at x where the edges are bought by the node x. You can see the topology in Figure 11. We claim that for $\alpha = 2n/5$ this graph is a NE.*

Proof. To begin with, notice that H is vertex-transitive and that the directed degree is at most two for every node in H. Therefore, in order to prove that G is a NE we only need to separate two cases: for every $u \in H$, u doesn't have an incentive of changing his/her strategy nor for every u and every $u' \in T(u)$, u' is a "son" of u if $|T(u)| > 1$. Consider all possible deviations that u can perform. We distinguish three distinct classes of deviations:

- $|S'_u| \geq |S_u|$. This deviation can be seen as at most two swaps with the addition of links. First it is clear that there cannot exist two bought links in the deviated configurations pointing to the same node in H. since H has diameter 2 then we clearly have that $\Delta C = k\alpha - kn/10 + l(n/10 - n/10) = k(\alpha - n/10) = k(3n/10) \leq 0$ where k is the number of

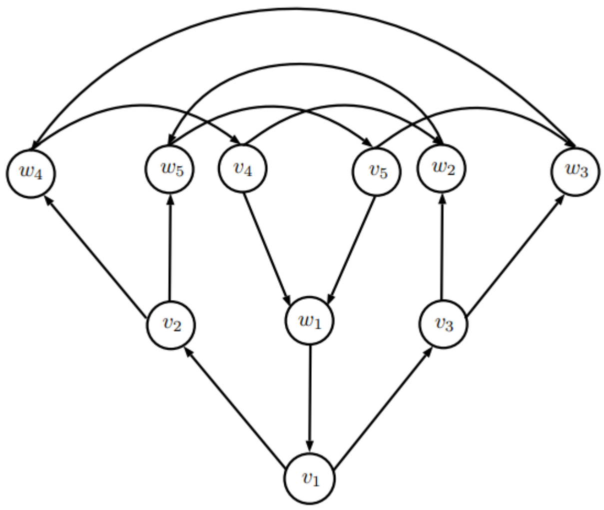


Figure 11: Layer visualization of the Petersen graph

extra bought links and l is the number of swapped links. We can easily see that this set of deviations brings no improvement.

- $|S'_u| < |S_u|$. For this set we need to consider two sub-cases:
  - $S'_u \subseteq |S_u|$. Then we only need to check that the cost difference associated to the deviation that consists in deleting one link has positive value. The corresponding cost difference ΔC_{delete} when u deletes one of its links is:

$$\Delta C_{delete} = -\alpha + 3n/10 + n/10 + n/10 = -2n/5 + n/2 = n/10 > 0$$

  - Otherwise, we must consider the deviations that consist in deleting plus swapping. Since u has bought at most two links we have to consider only the deviation that consists in deleting one link and swapping the other one which ends up having the same inequality as the previous sub-case.

$$\Delta C = \alpha - (2n/10 + n/10 + n/10) = 2n/5 + 2n/5 = 0$$

□

6 Greedy Equilibria

In this section we introduce new definitions of Equilibria in order to study the networks designed in the previous chapter under different conditions. One of the equilibria we use is the Greedy Equilibrium defined in [7] and [9] as follows:

We consider that each time the agents can improve their current strategy is done by selecting one, and only one, of the following three options:

- **Add**: the player buys one new link.
- **Erase**: the player remove one owned link.
- **Swap**: the player swaps one owned link for another not owned link.

Therefore we say that G is in Greedy Equilibrium (GE) when no agent can improve his/her cost function by buying, selling or swapping one owned edge. Here are some reasons to use GE stated in [7] and [9] that we find interesting when we try to shed light in the Tree Conjecture.

- **Computational Improvements**: Computing the greedy-best strategy for one agent can be done in $O(n^2(n+m))$, for n being the network size and m being the number of edges of said network. Although in this section we do not make an exhaustive use of this property it certainly is useful to improve the computational time that the *Best Response* needs.
- **Tree Sum-GE, then Sum-NE**: For the Sum version of this game, defined in 3.2, if G is tree and in GE, then G is NE. Although in this project we are focused on non-tree NE topologies, this property ensures us that those tree GE topologies explored are, in fact, also NE.
- **NE 3-approximate**: With Theorem 4 in [7] the authors prove that a GE is at most three times worst than a NE for the sum model. Therefore, by exploring this new way of attacking the problem we still have some rough results that could help us with the *Tree Conjecture* and the *Constant PoA Conjecture*.

Now we decide to split the Greedy Equilibrium into three different Equilibria because we want to analyze the behaviour of the topologies in as many ways as possible. Therefore, we can define the Add, Erase and Swap equilibria as:

- **Add Equilibrium:** No agent can improve his/her cost function by adding a new edge.
- **Erase Equilibrium:** No agent can improve his/her cost function by erasing an owned edge.
- **Swap Equilibrium:** No agent can improve his/her cost function by swaping an owned edge for a new one.

Hence, to evaluate if the network is in Add/Erase/Swap Equilibrium we make use of computer algorithms. Despite the fact that the previous definitions of equilibria imply a direction in the graph induced by the network, the algorithms that we present take as an input an undirected graph and the α value. The aforementioned algorithms return *true* if there is a possible direction of the edges such as the resulting directed graph is in equilibrium. The algorithms return $false$ if all directions are not in equilibrium. Here we present the preudocode of the algorithms we use.

Add Equilibrium: We use Algorithm 5 to evaluate if a certain topology is Add Equilibrium. This algorithm is, what we call, node-based, meaning that the algorithm iterates through all the nodes in the graph searching for a conflicting node. A conflicting node is any player that can improve his/her current strategy. A network is in Add Equilibrium if and only if there are no conflicting nodes. Note that the direction of the graph is not needed to compute if a certain player can improve his/her strategy by adding one new link. The reason is that we know the number of bought links will be increased exactly by one and we compute the distance towards the rest of the players with the undirected graph.

Algorithm 5 Add Equilibrium

```
function IS_ADD_EQUILIBRIUM(G, α)                                          ▷
    n = G.nodes().size()
    for i = 0; i < n; i++ do
        if not is_add_one_response(G, α, i) then    ▷ Returns true if the player cannot improve
his/her strategy, false otherwise
            return false
        end if
    end for
    return true
end function
```

Erase Equilibrium: We use Algorithm 6 to evaluate if a certain graph is Erase Equilibrium. This algorithm, along-side Swap Equilibrium algorithm, is edge-based. Edge-based means that we iterate through all the edges of the graph looking for conflicting edges. A conflicting edge, $e = (u, v)$, means that whoever the owner of this edge is, being it u or v, both of them will erase it o swap it to minimize their cost function. Therefore, a network is in Erase/Swap Equilibrium if and only if there are no conflicting edges.

In this case we consider the option of erasing edges and, before continuing, let us remember that the cost function for the sum model is:

$$c_u(s) = \begin{cases} \infty, & \textit{if } G[S] \textit{ is not connected} \\ \alpha|s_u| + \sum_{v \neq u} d_{G[S]}(u, v), & \text{otherwise} \end{cases}$$

To compute the cost difference of erasing one edge we have to calculate $c_u(s) - c_u(s')$ where, u is the player, s is the strategy before erasing the edge and s' is the strategy after erasing it. We know that $c_u(s) - c_u(s') = -\alpha + distance\_after - distance\_before$, where the distance after erasing the edge could be infinity if the induced graph has more than one connected component.

As it occurs with Algorithm 5, this algorithm receives as an input an undirected graph because the direction of the edges is not needed to compute if a player can improve his/her current strategy. Note that in all possible directed graphs every edge (u, v) can have one out of two directions, it is either bought by player u or by player v. Algorithm 6 checks both possible directions for every edge.

Algorithm 6 Erase Equilibrium

```
function IS_ERASE_EQUILIBRIUM(G, α)                                         ▷
    E = G.edges()                          ▷ Returns the list of all the edges belonging to G
    for each e in E do
        if erases_edge(G, α, e, e[0]) and erases_edge(G, α, e, e[1]) then  ▷ Do both players want
to eliminate edge e?
            return false
        end if
    end for
    return true
end function
function ERASES_EDGE(G, α, e, player)
    before_distance = calculate_distance(G, player)
    G.remove_edge(e)
    after_distance = calculate_distance(G, player)
    return before_distance > after_distance − α
end function
```

Swap Equilibrium: We use Algorithm 7 to evaluate if a given graph is Swap Equilibrium. This algorithm is edge-based, like Algorithm 6. But in this case we need to compute, like in Algorithm 5, all possible destination for the swappable edge. The reasons for having as an input an undirected graph are the same as in Algorithm 5 and 6. A special point to notice for this particular algorithm is that we do not take into account the α value since for every player the number of edges remains the same.

In the following sections we analyze the topologies shown in Figure 9 and Figure 10 with the help of the algorithms described above. For the Erase and Add Equilibrium we give an interval for each topology such as when the α parameter is inside the interval the topology is in equilibrium. To find those intervals we use a simple binary search algorithm in which we search the threshold for α value that results in an equilibrium between $[1, 4n - 13]$. As we all know, for the binary search we have a comparison that tell us if we have to keep looking in the left or the right part of the sorted interval or if we have found the item we were looking for. In our implementation we have to differentiate the comparison between the Add and Erase Equilibrium.

For the Add Equilibrium we know that if a certain topology is in AE for a given α value it means that the players do not have the incentive to buy new edges, meaning that the price per link is high. Therefore we have to search the left part of the interval since the α value will be smaller, meaning that the price per link will be cheaper. When we have that a certain topology is not in

Algorithm 7 Swap Equilibrium

```
function IS_SWAP_EQUILIBRIUM(G)                                              ▷
    E = G.edges()
    for each e in E do
        if swaps_edge(G, e, e[0]) and swaps_edge(G, e, e[1]) then
            return false
        end if
    end for
    return true
end function
function SWAPS_EDGE(G, e, player)
    before_distance = calculate_distance(G, player)
    G.remove_edge(e)
    dest_list = G.nodes()\{e[0], e[1]}\player.neighbours()    ▷ The destination list is the set of
available players to swap the edge to
    strategies = all_distances_sorted(G, player, dest_list, before_distance)  ▷
    return before_distance > strategies[0]                                   ▷ (1)
end function
```

AE we know that the price per link is too low and we have to increased, searching the right part of the interval.

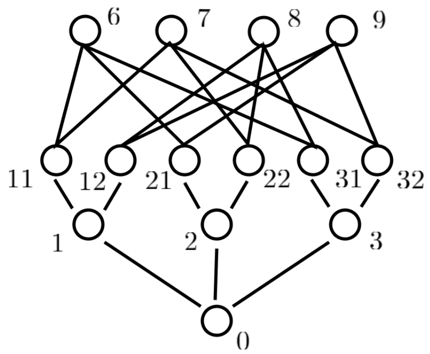
For the Erase Equilibrium we have the exact opposite as described for the Add Equilibrium. If we know that a certain topology is in EE for a given α value it means that the players do not have the incentive to sell owned links, meaning that the price per link is too low, and we have to search the right part. If the topology is not in EE we have to decrease the α value.

For the Swap Equilibrium, since Algorithm 7 does not require the α value, we give a counter-example if the topology is not in equilibrium.

In this study we also make use of *Node Weights* like in the previous chapter. We contemplate equal *Node Weights* for all the players, represented by the letter W and the topologies of the *Node Weights* are restricted to be stars where the central node is the one belonging to the biconnected component H and all the edges are bought by the central node. Since every node in H has equal weights we can define the size of G as $n = W|V(H)|$.

Finally, for the proofs we have a specific nomenclature. The cost difference is always with capital C, C, and it has a subindex pointing the player (for hanging nodes that belong in the *Node Weight* of a player we denote it with the letter h). The images to the sides are the pictured deviations where erased edges are denoted by dotted lines and new bought edges are denoted by dashed lines with an arrowhead for the direction.

Cinc-Comb



The topology Cinc-Comb is a result of adding a third layer to the Petersen graph in order to increase the diameter of this topology to 3. To do that we calculate how many nodes are needed in the last layer. Because of Lemmas 4 and 3 we know that the nodes in layer 2 and 3 must have degree of at least 3. Therefore we have the following equation: $2 * 6 = 3 * x$. Each node in layer 2 has two edges to connect to layer 3, and the nodes in layer 3 have 3 edges each. If we solve this equation we get that 4 is the number of nodes in the third layer.

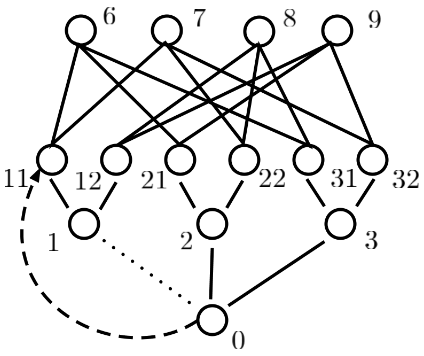
Once we have the number of edges and the number of nodes decided we need to connect the nodes from the second to the third layer and make sure the diameter of the 2-vertex-connected component is 3. Taking into account that there can't be any squares or triangles, Lemma 1, we have the following combination of edges, from left to right the nodes of the third layer have to connect to the nodes of the second layer: (1,1,1), (1,2,2), (2,2,1),(2,1,2). Where the numbers 1 or 2 serve to indicate if we take the left (1) or right (2) node of the second layer. Finally we know that for this topology $n = 14W$, where $W = \frac{n}{14}$ is the size of the *Node Weights*.

Results: We know that Cinc-Comb is not a Swap Equilibrium (SE), Cinc-Comb is an Add Equilibrium (AE) when $\alpha \geq \frac{9}{14}n$ and when $\alpha \leq \frac{8}{14}n$ Cinc-Comb is an Erase Equilibrium (EE).

Even though Algorithm 7 shows us that Cinc-Comb is not a Swap Equilibrium let us show a conflicting edge of the graph.

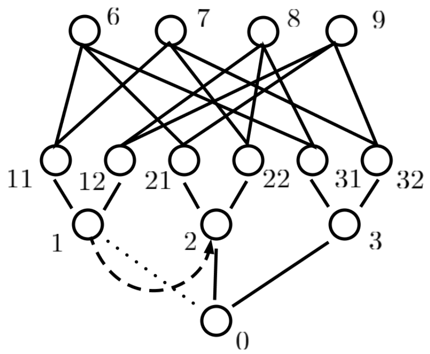
Proposition 4. *The Cinc-Comb graph is not Swap Equilibrium.*

Proof. Let us take the node 0 and swap the edge (0,1) for the edge (0,11). This results in a cost difference of $-W$ because: the number of bought edges remains the same (0), the distance towards 1 and 12 is increased by 1 ($2W$) and the distance towards 11, 6 and 7 is decreased by 1 ($-3W$).



$$\Delta C_0 = 0 \times \alpha + 2W - 3W = -W < 0$$

Let us take node 1 and swap the edge (1,0) for the edge (1,2). This results in a cost difference of $-W$ because: the node has the same amount of links bought (0), the distance towards 0 and 3 is increased by 1 ($2W$) and the distance towards 2, 21 and 22 is decreased by 1 ($-3W$).

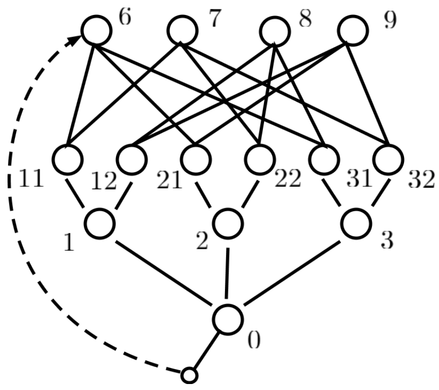


$$\Delta C_1 = 0 \times \alpha + 2W - 3W = -W < 0$$

□

Proposition 5. *Cinc-Comb is an Add Equilibrium if and only if $\alpha \geq 9W$.*

Proof. Since Algorithm 5 shows us that Cinc-Comb is AE when $\alpha \geq 9W$, let us consider Cinc-Comb when $\alpha < 9W$ and prove that there is a conflicting node. Let us take a hanging node from the node 0 and suppose this node buys a link towards Layer 3, for example to node 6. This results in a a cost difference of $\alpha - 9W$ because: a new link has been bought (α), the distance towards 6 is decreased by 3 ($3W$) and towards 11, 21, 31, 7, 8 and 9 is increased by 1 ($6W$). Since $\alpha < 9W$ we have that the node hanging from player 0 has incentive to buy a link towards node 6.

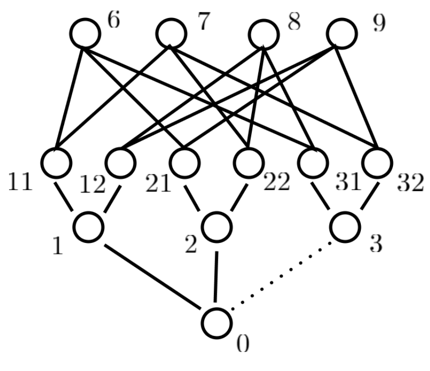


$\Delta C_{h_0} = 1 \times \alpha - 9W = \alpha - 9W < 0 \rightarrow \alpha \geq 9W$

□

Proposition 6. *Cinc-Comb is Erase Equilibrium if and only if* $\alpha \leq 8W$*.*

Proof. Since Algorithm 6 shows us that Cinc-Comb is EE when $\alpha \leq 8W$, let us consider Cinc-Comb when $\alpha > 8W$ and prove that edge $(0, 3)$ is a conflicting edge. On the one hand, if player 0 erases the edge, it results in a cost difference of $-\alpha + 8W$ because: the number of links has decreased by 1 ($-\alpha$), the distance towards 3 is increased by 4 ($4W$) and towards 31 and 32 is increased by 2 ($4W$). On the other hand, if player 3 erases the edge, it results in a cost difference of $-\alpha + 8W$ because: the number of links has decreased by 1 ($-\alpha$), the distance towards 0 is increased by 4 ($4W$) and towards 1 and 2 is increased by 2 ($4W$). Since $\alpha > 8W$ we have that the both players, 0 and 3, have incentive to erase the edge.

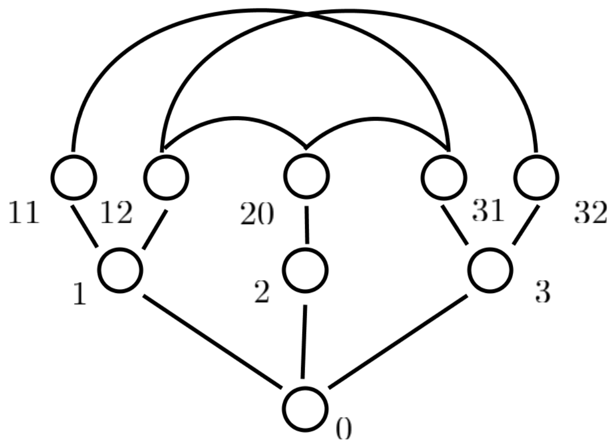


$\Delta C_0 = -1 \times \alpha + 8W = -\alpha + 8W < 0 \rightarrow \alpha \leq 8W$

□

Conclusions: As we can see, this topology is not a GE for any α value because there is a conflicting edge in the Swap Equilibrium. Nonetheless we have found two intervals in which Cinc-Comb is Add Equilibrium and Erase Equilibrium. We have found a topology with an equilibrium when $\alpha > n > 9W$.

Arnau-Cinc

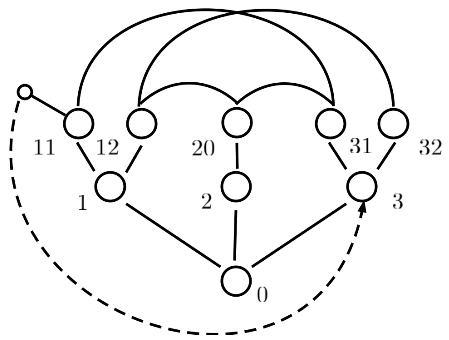


This topology is inspired by reducing the Petersen graph by a single node and re-arrange the number of edges. This change brings the diameter of the hole 2-vertex-connected component to be 3, but we can find nodes which have a local diameter of 2. For this topology $n = 9W$, where $W = \frac{n}{9}$ is size of the *Node Weights*.

Results: We know that Arnau-Cinc is SE because of Algorithm 7, Arnau-Cinc is AE when $\alpha \geq \frac{6}{9}n$ and when $\alpha \leq \frac{5}{9}n$ Arnau-Cinc is EE.

Proposition 7. *Arnau-Cinc is an Add Equilibrium iff* $\alpha \geq 6W$*.*

Proof. Since Algorithm 5 shows us Arnau-Cinc is AE when $\alpha \geq 6W$, let us consider Arnau-Cinc when $\alpha < 6W$ and prove that there is a conflicting node. Let us take a hanging node from the node 11 and suppose this node buys a link towards 3 resulting in a cost difference of $\alpha - 6W$. This is because the number of links increases by 1 (α), the distance towards 3 and 31 is decreased by 2 ($4W$) and towards 0 and 2 is decreased by 1 ($2W$). Since we know that $\alpha < 6W$ we can see that this node is, in fact, a conflicting node.

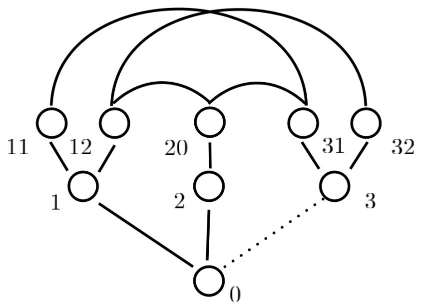


$$\Delta C_{h_{11}} = 1 \times \alpha - 6W = \alpha - 6W < 0 \rightarrow \alpha \geq 6W$$

□

Proposition 8. *Arnau-Cinc is an Erase Equilibrium iff* $\alpha \leq 5W$*.*

Proof. Since Algorithm 6 shows us that Arnau-Cinc is EE when $\alpha \leq 5W$, let us consider Arnau-Cinc when when $\alpha > 5W$ and prove that edge $(0,3)$ is a conflicting edge. On the one hand, let us take the node 0 and supposes he/she erases the edge with a cost difference of $-\alpha + 5W$ because the number of links is decreased by 1 ($-\alpha$), the distance towards 3 is increased by 3 ($3W$) and towards 31 and 32 is increased by 1 ($2W$). On the ther hand, let us take the node 3. After erasing the edge we have a cost difference of $-\alpha + 5W$ because the number of links is decreased by 1 ($-\alpha$), the distance towards 0 is increased by 3 ($3W$) and towards 1 and 2 is increased by 1 ($2W$). We can see that edge $(0,3)$ is a conflicting edges because $\alpha > 5W$.

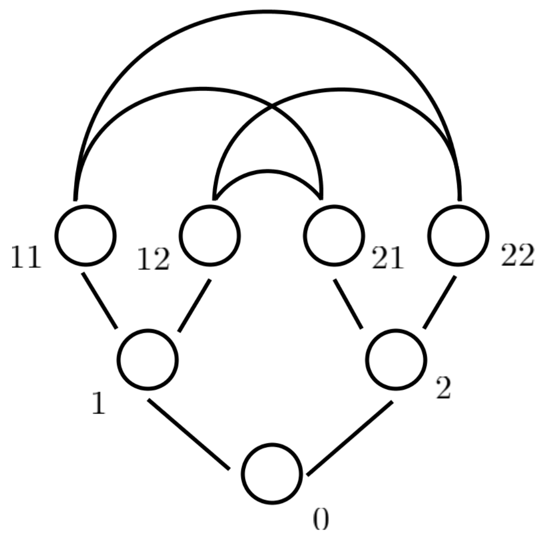


$$\Delta C_0 = -1 \times \alpha + 5W = -\alpha + 5W < 0 \rightarrow \alpha \leq 5W$$

□

Conclusions: In this case we have our first Swap Equilibrium, which indicates that the distances and number of edges are well balanced between the nodes. As for the Add and Erase configuration note that the intervals do not intersect. This means that in order to comply with the Add configuration the α value will cause the topology to not comply with the Erase configuration and vice-versa. Consequently this topology isn't GE. For the following topologies we will look if we get more Swap Equilibria or if the intervals intersect.

Petit-Cinc

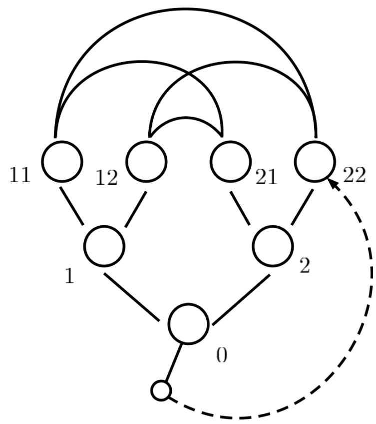


This is another designed inspired in reducing the Petersen graph. This time we erased a whole branch, 3 nodes and their edges. We try to preserve the overall degree of 3, except for the "base" node, and the diameter of the 2-vertex-connected component, which is still 2. We know this particular topology has squares but since now we are analyzing it with smaller α values we believe it is an interesting topology to consider. For this topology $n = 7W$, where $W = \frac{n}{7}$ is size of the *Node Weights*.

Results: After the algorithms we know that Petit-Cinc is SE, Petit-Cinc is AE when $\alpha \geq \frac{4}{7}n$ and when $\alpha \leq \frac{2}{7}n$ Petit-Cinc is EE.

Proposition 9. *Petit-Cinc is an Add Equilibrium iff $\alpha \geq 4W$.*

Proof. Since Algorithm 5 shows us that Petit-Cinc is AE when $\alpha \geq 4W$, let us consider Petit-Cinc when $\alpha < 4W$ and prove that there is a conflicting node. Let us take a hanging node from the node 0 that buys a link towards the node 22. This results in a cost difference of $\alpha - 4W$ because the number of links is increased by 1 (α), the distance towards 22 is decreased by 2 and towards 11 and 12 is decreased by 1 ($4W$). Because $\alpha < 4W$ we can see that the hanging node is a conflicting node.

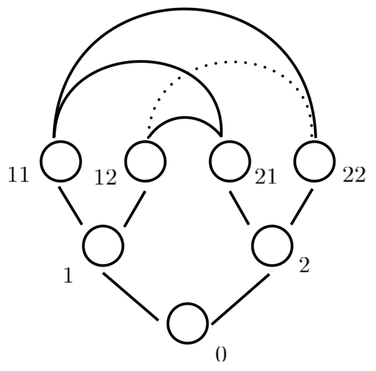


$$\Delta C_{h_0} = 1 \times \alpha - 4W = \alpha - 4W < 0 \rightarrow \alpha \geq 4W$$

□

Proposition 10. *Petit-Cinc is an Erase Equilibrium iff $\alpha \leq 2W$.*

Proof. Since Algorithm 6 shows us that Petit-Cinc is an EE when $\alpha \leq 2W$, let us consider Petit-Cinc when $\alpha > 2W$ and prove that edge (12,22) is a conflicting edge. On the one hand, let us take node 12 and erase the edge with a cost difference of $-\alpha + 2W$. That is because the number of links is decreased by 1 ($-\alpha$), the distance towards 22 is increased by 2 ($2W$). On the other hand, let us take node 22 and erase the edge with a cost difference of $-\alpha + 2W$. That is because the number of links is decreased by 1 ($-\alpha$), the distance towards 12 is increased by 2 ($2W$). As a consequence of $\alpha > 2W$ we know that edge $(12, 22)$ is a conflicting edge.

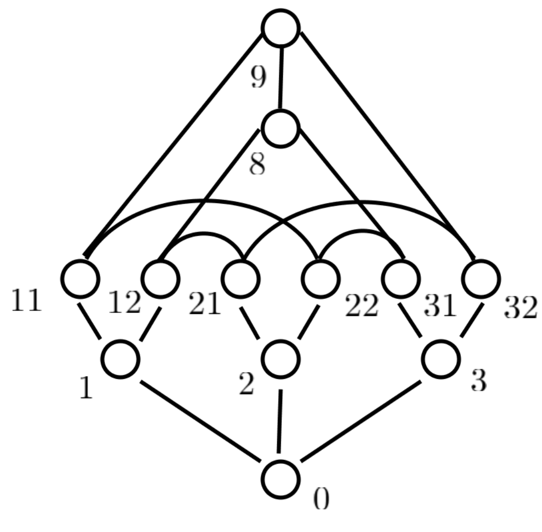


$$\Delta C_2 = -1 \times \alpha + 2W = -\alpha + 2W < 0 \rightarrow \alpha \leq 2W$$

□

Conclusions: In this case we have another Swap Equilibrium, which as before shows the balance between the distances and edges. As for the Add and Erase configuration unfortunately we reach the same conclusion as other cases, the intervals intersect and the topology is not a GE. With two topologies with similar characteristics we believe it will be profitable to investigate further in future work.

Sis-Sis

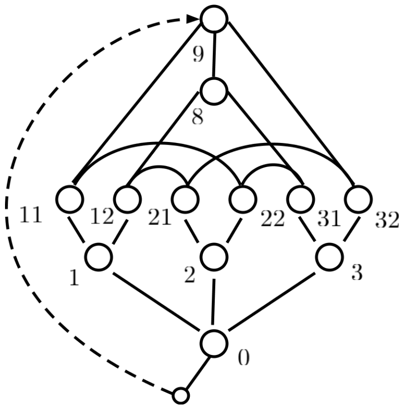


We came up with this topology after some experimentation. The idea behind it was to have a very small amount of nodes in the last layer since in the previous cases we have testes we did not look for it. Visually speaking this topology can be seen as two separate hexagons, one inside the other, with the nodes connected very carefully to not create squares or triangles. For this topology $n = 12W$, where $W = \frac{n}{12}$ is size of the *Node Weights*.

Results: We know that Sis-Sis is a Swap Equilibrium and the intervals for AE and EE are: when $\alpha \geq \frac{7}{12}n$ Sis-Sis is in AE and when $\alpha \leq \frac{5}{12}n$ Sis-Sis is in EE.

Proposition 11. *Sis-Sis is an Add Equilibrium iff $\alpha \geq 7W$.*

Proof. Since Algorithm 5 shows us that Sis-Sis is an AE when $\alpha \geq 7W$, let us consider Sis-Sis when $\alpha < 7W$ and prove that there is a conflicting node. Let us take a hanging node from 0 and buy a link towards 9, with a cost difference of $\alpha - 7W$. That is because the number of links is increased by 1 (α), the distance towards 9 is decreased by 3, towards 8 is decreased by 2 and towards 11 and 22 is decreased by 1 ($7W$). As we know that $\alpha < 7W$ we know that the hanging node has incentive to buy the specified link.

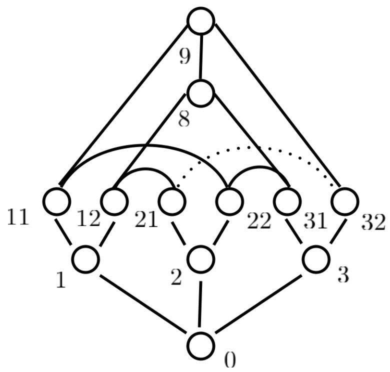


$$\Delta C_{h_0} = 1 \times \alpha - 7W = \alpha - 7W < 0 \to \alpha \geq 7W$$

□

Proposition 12. *Sis-Sis is an Erase Equilibrium iff $\alpha \leq 5W$.*

Proof. Since Algorithm 6 shows us that Sis-Sis is an EE when $\alpha \leq 5W$, let us consider Sis-Sis when $\alpha > 5W$ and let us prove that edge (21,32) is a conflicting edge. On the one hand, let us take node 21 and let us erase the edge towards 32. This results in a cost difference of $-\alpha + 5W$ because the number of links is decreased by 1 ($-\alpha$), the distance towards 32 is increased by 3 and towards 3 and 9 is increased by 1 ($5W$). On the other hand, let us take node 32 that erases the edge. This results in a cost difference of $-\alpha + 5W$ because the number of links is decreased by 1 ($-\alpha$), the distance towards 21 is increased by 3 and towards 2 and 12 is increased by 1 ($5W$). Lastly, knowing that $\alpha > 5W$ we have that edge (21,32) is indeed a conflicting edge.

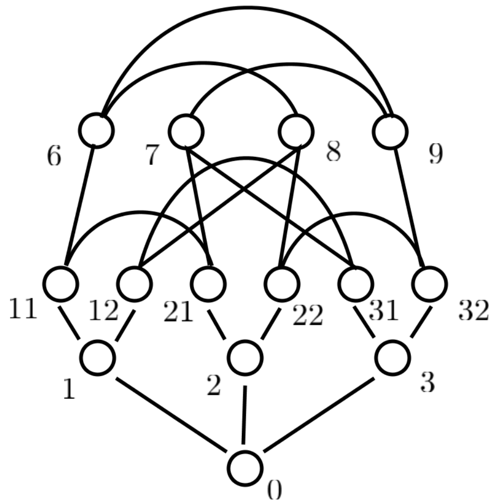


$$\Delta C_2 = -1 \times \alpha + 5W = -\alpha + 5W < 0 \to \alpha \leq 5W$$

□

Conclusions: We have found yet another topology that is Swap Equilibrium. We believe that this is an indication that the topological design is productive. Unfortunately the Add and Erase Equilibrium intervals intersect once again.

Set-Cinc



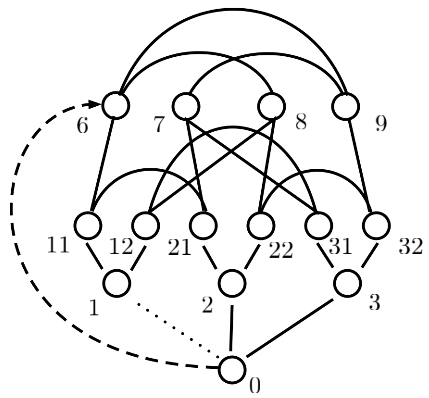
Last but not least, we have this design which can be seen as a heptagon inside another heptagon and, as the previous topology, the nodes are connected from the inside to the outside cycle carefully avoiding the formation of squares and triangles. This idea was very simple since we intended to maintain similar degrees between the nodes and an easy and fast way to do so was to have a cycle with an extra edge per node. For this topology $n = 14W$, where $W = \frac{n}{14}$ is size of the *Node Weights*.

Results: Set-Cinc is not a SE, Set-Cinc is an AE when $\alpha \geq \frac{9}{14}n$ and when $\alpha \leq \frac{5}{14}n$ Set-Cinc is an EE.

Even though Algorithm 7 shows us that Set-Cinc is not a Swap Equilibrium let us show a conflicting edge of the graph.

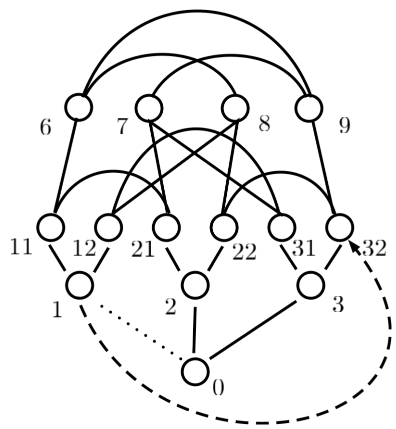
Proposition 13. *Set-Cinc graph is not a Swap Equilibrium.*

Proof. Let us take the node 0 and let us swap the edge (0,1) for the edge (0,6) resulting in a cost difference of $-W$. The number of links remains the same (0), the distance towards 1 is increased by 2 and towards 12 is increased by 1 ($3W$) but the distance towards 6 is decreased by 2 and towards 8 and 9 is decreased by 1 ($-4W$).



$\Delta C_0 = 0 \times \alpha + 3W - 4W = -W < 0$

Let us take the node 1 and swap the edge (1,0) for the edge (1,32) with a cost difference of $-W$. The number of links remains the same (0), the distance towards 0 is increased by 2 and towards 2 by 1 ($3W$) but the distance towards 32 is decreased by 2 and towards 9 and 22 is decreased by 1 ($-4W$).

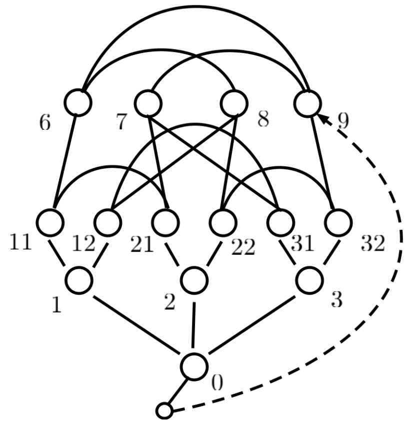


$\Delta C_1 = 0 \times \alpha + 3W - 4W = -W < 0$

□

Proposition 14. *Set-Cinc is an Add Equilibrium iff* $\alpha \geq 9W$.

Proof. Since Algorithm 5 shows us that Set-Cinc is an AE when $\alpha \geq 9W$, let us consider Set-Cinc when $\alpha < 9W$ and prove that there is a conflicting node. Let us take a hanging node from 0 that buys a link towards Layer 3, for example the node 9, with a cost difference of $\alpha - 9W$. This is due to the number of links increasing by 1 (α), the distance towards 9 decreasing by 3, towards 6 and 7 is decreasing by 2 and towards 8 and 12 decreasing by 1 ($9W$). We have that the hanging node is a conflicting node because, as we know, $\alpha < 9W$.

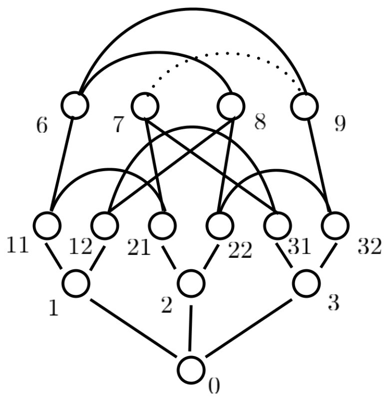


$\Delta C_{h_0} = 1 \times \alpha - 9W = \alpha - 9W < 0 \rightarrow \alpha \geq 9W$

□

Proposition 15. *Set-Cinc is an Erase Equilibrium iff* $\alpha \leq 5W$*.*

Proof. Since Algorithm 6 shows us that Set-Cinc is an EE when $\alpha > 5W$, let us consider Set-Cinc when $\alpha > 5W$ and prove that the edge (7,9) is a conflicting edge. On the one hand, let us take the node 9 and let us erase the edge towards 7. This results in a cost difference of $-\alpha + 5W$ because the number of links is decreased by 1 ($-\alpha$), the distance towards 7 is increased by 3 and towards 21 and 21 is increased by 1 ($5W$). On the other hand, let us take node 7 and erase the aforementioned edge. This results in a cost difference of $-\alpha + 5W$ because the number of links is decreased by 1 ($-\alpha$), the distance towards 7 is increased by 3 and towards 21 and 21 is increased by 1 ($5W$). Because $\alpha > 5W$ we know that edge (7,9) is a conflicting edge.



$\Delta C_0 = -1 \times \alpha + 5W = -\alpha + 5W < 0 \rightarrow \alpha \leq 5W$

□

Conclusions: With a topology with that many edges it is no surprise that Set-Cinc is not Swap Equilibrium. What interests us the most is that from all the other topologies, Set-Cinc has the biggest distance between the intervals for AE, when $\alpha \geq 9W$, and EE, when $\alpha \leq 5W$.

Conclusions on Different Equilibria

To conclude this last analysis we have found some interesting results. To begin with, Sis-Sis, Petit-Cinc and Arnau-Cinc are Swap Equilibria and this could be because of how the topology is connected. It is interesting to know if there is some property implying that the topologies are in Swap Equilibrium and it is material for future work.

Next, Cinc-Comb and Arnau-Cinc have relatively near intervals of Add and Erase Equilibrium. They are the two topologies with the most similar structure to the Petersen graph. We believe that in future work it would be profitable to study new properties in order to approach those two intervals even closer.

Finally, for the rest of the topologies we have very distant intervals of AE and EE, like in Set-Cinc where Add and Erase intervals differ for at least $4W = \frac{4}{14}n$. We believe it is due to the amount of edges and the actual position but we consider to study other possible implications and try to gather some new properties.

7 Conclusions and Future Work

In this chapter we summarize all the conclusions and results that we have gathered with the different experiments and analysis conducted throughout this project.

First, with the first experiments we wanted to get a wide view of the problem from an empirical standpoint. Even though we anticipated some computational problems we encountered barriers that we could not solve and had to evade, one of those barriers was the network size. Therefore, for the second set of experiments we decided to somehow restrict the topologies generated. As a result from the first experiment we now know that there are no NE networks when $\alpha = n$ and $n \in \{3, 4, 5\}$.

Then, for the second run of experiments we began by shrinking the search space and increasing the number of topologies that we analyze compared to the previous experiment. Despite searching for topologies with specific properties, it was not enough to generate a sufficient amount of networks for the purpose of conducting a more specific analysis. In our efforts to increase the number of topologies to work on with the minimum generations possible we decided to use *Node Weights* and design networks manually for the following topological analysis. For this second run of experiments we have a big amount of generated topologies when $n \in \{10, 15\}$ and we did not find any non-tree NE graps.

Next, after analyzing the topologies seen in Figure 12 and 13 we found a NE graph when $\alpha = \frac{2n}{5}$. With this new finding we increase the number of known NE graphs when the α value is smaller but close to n, some of the NE graphs can be seen in Figure 2 and Figure 3.

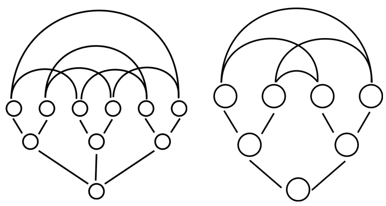


Figure 12: Manually designed topologies of diameter 2.

To our surprise this topology is the Petersen graph. Since this specific graph has been studied a lot we believe that in future work we could go through all the properties found and try to find new properties that we could use to generate other NE networks.

After that, we decided to relax the definition of equilibria that we had been using throughout the past three experiments. The new equilibria that we used to analyze our designed topologies is the Greedy Equilibrium. We knew that by using the GE we could apporximate a NE, as proved in [7] and [9]. This property ensured us that the outcome of this new experiment could help us draw conclusions on our main problems.

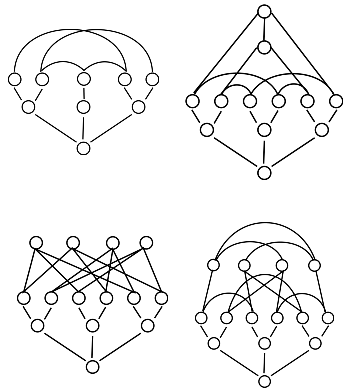


Figure 13: Manually designed topologies of diameter 3

Here we present a compact table of the different results that we found for our designed topologies.

Cinc-Comb

SWAP	ADD	ERASE
This topology is not **Swap Equilibrium** because of edge (0, 1).	When $\alpha \geq \frac{9}{14}n$ this topology is **Add Equilibrium**.	When $\alpha \leq \frac{8}{14}n$ this topology is **Erase Equilibrium**.

Arnau-Cinc

SWAP	ADD	ERASE
Exists a direction for this topology that is **Swap Equilibrium**	When $\alpha \geq \frac{6}{9}n$ this topology is **Add Equilibrium**.	When $\alpha \leq \frac{5}{9}n$ this topology is **Erase Equilibrium**.

Petit-Cinc

Exists a direction for this topology that is **Swap Equilibrium**	When $\alpha \geq \frac{4}{7}n$ this topology is **Add Equilibrium**.	When $\alpha \leq \frac{2}{7}n$ this topology is **Erase Equilibrium**.

Sis-Sis

SWAP	ADD	ERASE
Exists a direction for this topology that is **Swap Equilibrium**	When $\alpha \geq \frac{7}{12}n$ this topology is **Add Equilibrium**.	When $\alpha \leq \frac{5}{12}n$ this topology is **Erase Equilibrium**.

Set-Cinc

SWAP	ADD	ERASE
This topology is not **Swap Equilibrium** because of edge (0, 1).	When $\alpha \geq \frac{9}{14}n$ this topology is **Add Equilibrium**.	When $\alpha \leq \frac{5}{14}n$ this topology is **Erase Equilibrium**.

From all the experiments and studies we have gathered some relevant information. To begin with, a lot of the topologies studied above are Swap Equilibrium and this denotes a certain balance between between the different nodes in the network. Next, for all the topologies discussed in Chapter 6 we have found that when $\alpha > n$ they all are Add Equilibrium. Although Add Equilibrium is a weak equilibrium if compared to the NE, we think relevant properties and restrictions may be found in future work. It is important to notice that the critical equilibrium that we encountered in order to find a GE when $\alpha > n$ is the Erase Equilibrium, because we found it was easier to sell links when the α price was so expensive, bigger than n.

Finally, this results give us a little bit more information about NE topologies when α is close to n. We think that in future work we can improve the properties gathered in this project and even help us improve Proposition 1 to include those topologies with the H component of bigger diameter, for example 3 and 4.

References

[1] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. "On Nash equilibria for a network creation game". In: *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. 2006, pp. 89–98.

[2] C. Àlvarez and A. Messegué. "Network creation games: Structure vs anarchy". In: *arXiv preprint arXiv:1706.09132* (2017).

[3] C. Àlvarez and A. Messegué. "On the Constant Price of Anarchy Conjecture". In: *arXiv preprint arXiv:1809.08027* (2018).

[4] D. Bilò, K. Choudhary, L. Gualà, S. Leucci, M. Parter, and G. Proietti. "Efficient Oracles and Routing Schemes for Replacement Paths". In: *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*. 2018, 13:1–13:15. DOI: 10.4230/LIPIcs.STACS.2018.13. URL: https://doi.org/10.4230/LIPIcs.STACS.2018.13.

[5] E. D. Demaine, M. Hajiaghayi, H. Mahini, and M. Zadimoghaddam. "The price of anarchy in network creation games". In: *ACM Transactions on Algorithms (TALG)* 8.2 (2012), p. 13.

[6] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. "On a network creation game". In: *Proceedings of the twenty-second annual symposium on Principles of distributed computing*. 2003, pp. 347–351.

[7] P. Lenzner. "Greedy selfish network creation". In: *International Workshop on Internet and Network Economics*. 2012, pp. 142–155.

[8] P. Lenzner. "On dynamics in basic network creation games". In: *International Symposium on Algorithmic Game Theory*. Springer. 2011, pp. 254–265.

[9] P. Lenzner. "On selfish network creation". In: Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2014.

[10] A. Mamageishvili, M. Mihalák, and D. Müller. "Tree Nash equilibria in the network creation game". In: vol. 11. 4-5. Taylor & Francis, 2015, pp. 472–486.

[11] A. Messeguè. *Looking for small diameter non-tree NE graphs*. Manuscript.

[12] A. Messeguè. *Small diameter and high girth NE*. Manuscript.

[13] M. Mihalák and J. C. Schlegel. "The price of anarchy in network creation games is (mostly) constant". In: *International Symposium on Algorithmic Game Theory*. 2010, pp. 276–287.

Appendices

A Project Management

In this chapter we present all the information to evaluate the viability, sustainability and risks that his project has. Through-out different sub-sections we will expose and discuss all the decisions that we have made when planning this research.

A.1 Temporal planning

In this section the temporal planning of the hole project will be summarized and explained. This will serve also to expose the different resources needed in the development of this particular project and the tasks that form the distinct parts of it.

A.1.1 Tasks Description

To start I will give a summarize of the tasks that form the project and explain its importance.

Project Definition: The first task is to center the project in a specific field, ours is Algorithmic Game Theory. In here we also specify the problem we will be working on taking into account all the previous work done around our problem, Network Creation Games.

Once we have set our scope and the objectives are clear we have determinate how to distribute the time we have into workable and achievable milestones.

Tools and Resources: For this projects, since the code behind it won't be that big compared to other projects we choose Python as our language-code of choice. To do that we looked for simplicity, flexibility and the amount of packages and support online.

The rest of the software and hardware were chosen mainly due to accessibility, like a personal computer and python packages. Here I present the tools and resources that will be used in this whole project.

- Workspace (windows + bash)
- Documentation (LaTeX)
- Sharing and communication (gmail, dropbox, drive)
- Language (Anaconda2 + Python v.2.7.13)
- Network package (NetworkX v.1.11)
- Graphic visualization (python packages like PyPlot
- A computer (my personal laptop will suffice)

The principal packages featured in this project, and a brief description, are:

- <u>NetworkX:</u> Used for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.
- <u>Pyplot:</u> Its simplicity and utility make the choice over other network visualization libraries.
- <u>PDF Backed:</u> package to create and manipulate PDFs which be used to generate file of the relevant data to make readability easier.

Model Proposal and Study: The model we are interested in is the Network Creation Games, more precisely the sum-model. In a NCG each player can be define as a node in a graph and that is why we decide to use NetworkX as our main Python package.

For the posterior experimentation we want study the properties and behavior of the networks. We will talk about the Tree Conjecture, the Constant PoA Conjecture, the previous work done in this problem and new theoretical results.

More detailed information will be given in Chapter 3.

Experimentation and Result Analysis: We plan different experiments starting from a blind introduction and then use the data gathered to find properties we can use to guide our posterior experiments. Said experiments will be done with very strict restrictions on the price per edge because we want to extend the results afterwards. We expect to reach a "wall" in our experiments where we will need to switch the computer for theoretical study. Once this modification occurs we will relax our restrictions and introduce new definitions of equilibria. Therefore, we hope to have different results with bigger boundries.

Documentation: This last task is used to polish the results and the documentation of the final milestone. In this document will be written all the work done during the project development and it compromises the budget and sustainability study, the code used for the experiments and the conclusions and future work available.

Budget: In this section we will discuss and expose the economic impact of the project. We will divide them into direct costs, indirect costs and take into account possible problem we might run into in the future. With all the different costs explained we will then propose a budget.

A.1.2 Direct Costs

In order to analyze the impact of the project we will count the resources needed, including human resources, software and hardware. The following table shows the required resources in the development phase of the project, its initial and final date and the amount of hours of use (rough approximation).

The team needs to keep up the time schedule in order to not fall behind and the total amount of waste needs to be low, such as copies, scans or draft papers. Apart for that there is no more pruning to be done in the physical resources.

Table 3: Resources

Name	Initial date	Final Date	Work (in hours)
Project director	Mon 12/02/2018	Fra 22/06/2018	200
PhD Assistant	Mon 12/02/2018	Fra 01/06/2018	240
Programmer	Mon 19/02/2018	Fra 29/06/2018	750
Laptop	Mon 19/02/2018	Fra 29/06/2018	500

Table 4: Direct Cost Budget

Name	Price (€/hour)	Work (in hours)	Total (€)
Project director	40	200	8,000
PhD Assistant	35	240	8,400
Programmer	25	750	18,750
Laptop	-	-	850
Office Supplies	-	-	50
Total		36,050 €	

A.1.3 Indirect Costs

The project duration is estimated to be 4 months. In this time period the efforts will be fixed on the project development and the analysis of the results. In order to accomplish this two steps the resources needed are a laptop and a programmer for the development and paper, a whiteboard with the markers and the PhD Assistant for the analysis.

Table 5: Energy Consumption

Name	kWh	Work (in hours)	Total Energy (kWh)
Electricity	0.1	1000	100
HP LaserJet 4250	0.680	40	13.6
Laptop	0.04	600	24
Total		137.6 kWh	

The project energy consumption is 137.6 kWh. This is a low result, granting it a good mark on consumption design. This result is taking into account that the laptops and the printer are not connected while they are not required, minimizing the consumption of electricity and the electricity, which we take into account four fluorescent bulbs, are open only 8 hours per day.

With a simple search for the price of the electricity we get the following budget for the indirect costs:

Table 6: Indirect Costs Budget

Name	Total Energy (kWh)	Price (€/kWh)	Total
Electricity	137.6	0.237	32.65 €

As we can see, and as expected, the indirect costs are eclipsed by the direct costs.

A.1.4 Unplanned Costs

Some of the surprises we may come upon are mainly hardware repairs. Notice that on the direct costs we added the price for a new laptop, thus if an incident on this piece of hardware occurs we can replace or fix the broken computer swiftly and still be on budget. Because of that precaution we know that yes we inflated the direct costs budget a little bit but, we accomplish that this serious problem can be negligible in time and in unplanned expenses.

A.1.5 Final Budget

After a thoughtful analysis of the costs the final budget looks like this:

Table 7: Final Budget

Concept	Price
Direct Costs	36,050
Indirect Costs	32.65
Total	36,082.65 €

A.2 Sustainability

The goal of this report is to define the viability and sustainability of the project by using a specific methodology, the Sustainability Matrix. The analysis in this report will be divided in three parts: Project Development, Exploitation and Risks.

Keep in mind that my project is scientific research and the outcomes of it are demonstrations and results but not a proper software product, some parts of the analysis lack meaning since there is no exploitation of the "product". Nonetheless, the outcome of it can also be seen as universal, meaning that can be used by other researchers around the world with no cost associated to us or to them.

A.2.1 Matrix Ponderation

To start I will give the ponderation of the matrix using the following criteria described in the "Sustainability Report" of the FDP. The overall result will give a an idea of the sustainability of the project and its criteria is as follows :

- A score from 0 to 10 points for each of the cells corresponding to the Project Development.
- A Score from 0 to 20 points for each of the cells corresponding to the exploitation phase.
- A score from -20 to 0 for the cells corresponding to the risks.

Table 8: Sustainability Matrix

	Development	Exploitation	Risks
Environmental	5	20	-5
Economic	5	20	-15
Social	8	20	0
Range	18	60	-20
Total		58	

With a sustainability range of 58 points the project sits on a very sustainable position. The development phase is quite good but the points drop because of the possible hardware problems, the amount of paper used for scans and copies can increase and the amount of hours to be done impact the final budget and environmental footprint. The exploitation phase is the most sustainable since there is no lifespan maintenance because of the project produces no product, its goal is to improve the theoretic bases and those improvements and discoveries are much more lasting than other projects. The risks of this project are mostly economical, due to the fact that there is no guarantee that there will be relevant results that result in an economic compensation.

A.2.2 Economic

Since the majority of the final budget goes to human resources we can estimate this project to be very economically viable. But, as said before, is is not ranged as the most viable because of the uncertainty of the outcomes which could make this project, if seen in a lucrative point of view, not as appealing as other may.

A.2.3 Social

This project can contribute into a better understanding on network creation games, their stability and social costs. It may be applied to make networks between people more reliable and robust improving the quality of life of those individuals.

On a personal level this project will contribute on expanding the professional and technical level of the team members and consolidate their theoretic knowledge and, on a team level, it will contribute on improving communication and coordination skills among the team members. Finally the documentation of the project will improve the writing skills of the writer and his ability to communicate abstract and complicated definitions to others.

A.2.4 Exploitation

As said before, the project goals are not to create a product, it has no economical exploitation. The goal of this project is to conduct experimental an theoretical research on Game Theory. Therefore the exploitation of this project has a more symbolic and abstract tune. We could state that if the results of this project are actually applied to so many fields, that its exploitation life is potentially endless. Pushing the boundaries of the human knowledge is everlasting.