STOCKHOLM SCHOOL OF ECONOMICS
Department of Economics
5350 Master's thesis in economics
Academic year 2022-2023

NOBODY TAKES IT ALL

Analysis of dynamic propagation of knowledge in a social network

Filippo Allegri (41872) and Urja Jain (41885)

In this paper, we analyse how knowledge is shared in a social network, with the purpose of defining recurring patterns and equilibria. We develop a formal model, the Know-It-All game, based on the PageRank algorithm and other related game theoretical models, such as the Buck-holding game. The game provides a set-up where the players choose to either share their knowledge with the neighbours or to keep it to themselves. We observe that this simultaneous choice made by the players results in certain equilibrium conditions with stable knowledge-sharing loops. This analysis is then extended to large and complex networks through a machine learning algorithm. We find the effect of knowledge complexity and knowledge transmissibility on these sharing loops, complementing the existing literature on knowledge-sharing networks. The model converges on many states in which players cooperate to reach a common good without communication or retaliation. No player can obtain the most out of the game alone, therefore, Nobody Takes It All.

Keywords: Knoweldge, Network, Game theory, Machine Learning

JEL: D80

Supervisor: Karl Wärneryd
Date submitted: 05-12-2022
Date examined: 14-12-2022
Discussant: Yuanqing Dong
Examiner: Kelly Ragan

Acknowledgement

We would like to express our deepest gratitude to Olle Törnquist and Silvia Siri for their precious help and their stoic patience.

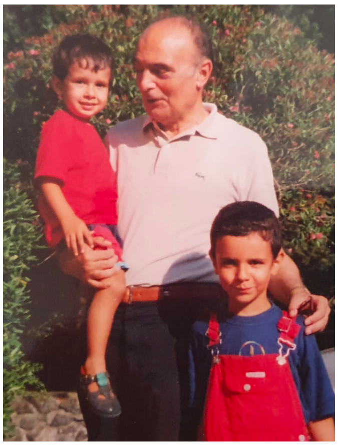
Moreover, Filippo would like to thank one person and he would like to do that in Italian.

Per Tito Siri:

Ciao Nonno,

Hai visto? Questa è la mia tesi di laurea. Molto di quello che c'è qui dentro viene dai momenti passati insieme a te: dalle partite a poker alle corse sotto i portici, dalle monete ai 12 gradi sul Beigua.

Grazie tanto per tutto
Filippo



Contents

List of Figures

List of Tables

1 Introduction

Nikola Tesla's alternating current, Guglielmo Marconi's Radio device, the wheel. These inventions are arguably among the most important in the history of humankind. They provide a basis for most of today's technology. The knowledge spread related to these inventions triggered a chain reaction where every new finding was based upon the previous one, leading to modern cars, computers, smartphones, and to the popular tv show "Wheel of fortune" (for which all of the three technologies mentioned in the beginning are necessary). Nevertheless, not every piece of knowledge is forever. What happened to Greek Fire and Damascus Steel? Why is no one able to fabricate a Stradivari violin today? It is clear that not all knowledge spreads in the same way, spreads at the same speed, or even spreads at all. Throughout this paper, we will use game theory and machine learning to model how knowledge spreads in social networks, finding equilibria and recurring patterns.

The literature recognizes the value of knowledge creation through the process of knowledge sharing within organizations and economies. Knowledge, considered as technological growth or human capital, is the driving input for economic growth (Romer 1993). But, as Friedrich Hayek (1945) points out, knowledge and its benefits can be maximized only by bringing together unique pieces that are spread across a wide distribution. On the other hand, Stauffer in his 1999 paper "Why people hoard knowledge" discusses knowledge hoarding as an innate need that arises out of the relative power that knowledge provides. Keeping this dichotomy in mind, knowledge sharing becomes a strategic choice. Many variables, like knowledge's nature and social ties, have been identified as affecting knowledge sharing by individuals (Cummings, 2003). Social network analysis offers an effective technique for understanding information-sharing networks (Haythornthwaite, 1996). More precisely, this technique, as well as strategic decision-making, will be considered and studied through our game theoretical model. We aim to bring together into one model separate studied aspects of knowledge sharing, such as relative knowledge distribution, knowledge-sharing networks, and the strategic choice of players to participate in this knowledge-sharing process.

The paper will introduce a new game-theoretical model, the Know-It-All (KIA) game. Consider the following set-up: players are connected to each other in a given network. Each player possesses a share of knowledge and decides to either share this knowledge with one of their connections or not. There is no direct cost of sharing this knowledge and players' knowledge increases when their neighbours decide to share knowledge with them. The question we seek to answer with this model is simple: "How is knowledge shared within the given network?". More generally, our Know-It-All game attempts to mimic how players share knowledge in a network where other players' actions define their payoff. The process that we use to determine this payoff is inspired by the PageRank mechanism.

To observe the strategies adopted by players in large and more complex networks, we use a machine learning algorithm that replicates the model. By showing the algorithm's fitness to the model for 3-player networks, we illustrate the converging strategies in more extensive and complex networks. These converging strategies point towards the existence of stable knowledge-sharing loops. We find that the size and the stability of these loops are positively related to the complexity and the transmissibility of the shared knowledge.

We start with the literature review, followed by an informal explanation of the functioning of the model. We will then discuss formal definitions, player strategies, and expected equilibria. Then, we present the simulation and its results. In the final sections, we discuss limitations and possible extensions of the model.

2 Literature Review

2.1 PageRank Literature

In order to understand how knowledge-sharing networks are formed, we took inspiration from the PageRank mechanism, the PageRank game and the buck-holding game. Therefore, we start by explaining the PageRank mechanism, its purpose, and its applications. We then discuss the papers highlighting the strategic decision-making aspect of this mechanism through a game-theoretic approach.

2.1.1 The PageRank Mechanism

In the 1998 paper "The Anatomy of a Large-Scale Hypertextual Web Search Engine", Sergey Brin and Lawrence Page presented Google and how this large-scale search engine will use hyperlinks to bring order to the numerous webpages available on the world wide web. The idea was to attach dynamic hierarchies to the websites based on the number and quality of links attached to them. The paper provides details on the PageRank algorithm to create this index and provides an objective value. This value is allotted such that a page can have a high PageRank under two conditions:

- Many pages point towards it through a hyperlink
- A page with high PageRank points towards this page

More understandably, this method provided a relative value – PageRank – to each page available on the internet based on its positioning in the network. This value is not only impacted by connections attached to the page but also by the value of its connections.

Looking at how this value is allocated to each web page, it can be understood that PageRank can not be ascertained through a single interaction, since the PageRank of each page is interdependent. However, it is necessary to converge to a stable value for the web pages to be indexed for Google search. To solve this issue, the PageRank algorithm uses iterative calculations. The idea is to count the value of in-links pointing to the webpage over several iterations, such that it eventually stabilizes to one value, which can be called its final PageRank.

This value, provided to each node of the web in the network, could be considered their "social value" (Pasquinelli, 2009). Interestingly, this was inspired by the academic citation system, which in itself is a system built to map knowledge creation.

2.1.2 The PageRank Game

Hopcroft and Sheldon (2008) describe the PageRank mechanism as a "link-based reputation measure" which can be dealt with as a network formation game, where players act strategically to increase their PageRank. The paper treats this strategic behaviour as a manipulation of the mechanism and attempts to understand its consequences. The paper finds equilibrium strategies in network reputation games where players maximize their reputation. In the model played on a directed graph, the players have no control over the in-links that they receive and the out-links are placed randomly. The important conclusions from the paper that are relevant to the thesis are as follows:

- The best response strategy for the player is to point towards the connections that are pointing towards the player
- The best response strategy set for players receiving no in-link is an empty set

2.1.3 The Buck-holding Game

Another game that has inspired our model looks at the PageRank game from an alternative perspective.

In the buck-holding game, described by Cominetti et al (2022), the players are passing a "buck" to other players in a network. The buck is provided randomly to a player at the beginning of the game, then it is passed around by the players aiming to maximize the probability of receiving it back. The players' strategy set is the set of players connected to them in the network. The paper describes the model theoretically with deterministic strategies, where single players play one strategy throughout the game, or in a stochastic manner, where the players assign probabilities to their strategies. The authors show that the game has multiple nash equilibria.

2.2 Knowledge Literature

Our question heavily depends on understanding knowledge, its properties and its definitions. This part of the literature review explores how knowledge is shared, how sharing impacts its value, and what happens to the resulting network.

2.2.1 Definition of Knowledge

The definition of knowledge has varied across disciplines and time. In economics, knowledge has often been interchangeably used with terms like technology and information.

In his book "The Economics of Knowledge", Dominique Foray (2004) discusses how knowledge and information differ from other tangible goods and require special attention regarding how their creation and distribution can be optimized. As the book points out, knowledge is a partially non-excludable and non-rivalrous good with a cumulative growth rate. Many papers like Lin (2007), Ohlsson (2011), and Foray (2004) have highlighted knowledge as an output and an essential input of itself. More precisely, knowledge is critical for generating and analyzing new information and innovation, leading to the creation of more knowledge. Furthermore, the quality of knowledge produced in this process also depends on the novelty of the combination and transfer of previous knowledge (Galunic & Rodan, 1998).

Antonelli (2005b, 2006) elaborates on how knowledge can be classified based on four major properties - level of tacitness, indivisibility, complementarity, and appropriability. A high level of tacitness implies increased difficulty in communicating knowledge with another agent. Knowledge indivisibility refers to how information can be divided and distributed among the agents. Complementarity highlights how knowledge differs from the existing one. Based on these features, how the knowledge is appropriated depends on the strategic conduct of the agents.

Due to the role of context and specialization, knowledge is often fragmented (Machlup, 1978). Therefore, gathering this dispersed knowledge becomes a fundamental economic problem, highlighting issues associated with centrality (Hayek, 1945). This centrality of knowledge distribution also emphasises its localized nature (Atkinson & Stiglitz, 1969).

2.2.2 Knowledge Sharing and Game Theory

Decisions and choices regarding how to share knowledge can be posed as a dilemma. Given that knowledge is a highly valued good, sharing it involves partially giving up the right to earn its benefits exclusively. Hence, it can be studied as a conflict between individual and group interests (Kimmerle et al., 2011). Sharing of knowledge can be perceived by agents as a loss of personal competitive advantage which can incentivize players to hold on to their share and preserve this advantage (Anand et al., 2020). Cabrera and Cabrera (2002) discuss the cooperation and public-good dilemma related to knowledge.

McLure-Wasko and Faraj (2005) discuss motivations for knowledge sharing in detail. In organizations, individuals prefer sharing knowledge for altruistic reasons like helping others or egoistic reasons, including improving personal reputation and influence. Lam and Lambermont-Ford (2010) and Hung et al. (2011) also suggest a combination of altruism and reciprocity drivers for knowledge sharing. It might also depend on the agent's position in the network. If the agent has a higher number of connections and a higher level of knowledge, their role is very consequential for knowledge sharing in the whole network.

Various theoretical and empirical studies have investigated knowledge sharing for research and development efforts. For example, Paier and Scherngell (2010) empirically study the network effect on R&D collaborations in Europe. The paper highlights that prior acquaintance, thematic proximity, and geographical proximity influence collaboration choices and the resulting network. The idea of a cooperative equilibrium, where firms share their R&D effort, is a plausible situation in certain conditions. Saint-Paul (2003) shows that this equilibrium is more likely when there are more firms and it is not as dependent on punishment strategies. The paper also argues for the cumulative nature of innovation and the higher rate of growth of innovation in the case of more cooperative equilibria. Yang and Wu (2008) study knowledge sharing in a firm by considering different player strategies - competitive, tit-for-tat and uncooperative. They conclude that the non-competitive strategy is the most dominant; however, the cooperative strategy can be persistent if the knowledge-sharing profits are high enough.

Networks based on interpersonal bonds influence how and why certain information and knowledge are shared most effectively (Hansen, 1999). Kelly and Grada (2000) empirically show how market panic spread through social networks depending on agents' place of origin and how these networks were the critical determinants of agents' behaviour. Additionally, Conley and Udry (2010) illustrate how helpful information about technology and its use spreads across friends and family networks. Chaudhary et al. (2016) studied knowledge sharing in hunter-gatherer communities based on the similarity of the medicinal plants they use. The study shows the impact of social structures based on family and kin relationships above the ecological variations.

Al-Gharaibeh and Ali (2020) review the different game theoretical approaches used in the literature to study knowledge-sharing frameworks. The paper shows the use of prisoner's dilemma games and assurance games (Fehr & Gachter, 2000) with repeated interactions to establish the dominant strategies and the conditions for convergence towards knowledge-sharing equilibria.

2.2.3 Network Theory and Knowledge Networks

Network theory mainly deals with the mechanisms and processes that form a network and how its structure can be defined. In a network, nodes are connected to each other through certain ties. The pattern in which these ties are related defines the network structure and the position of the different nodes (Borgatti & Halgin, 2011; Brass, 2002).

Along similar lines, knowledge networks are defined as a set of nodes. Each of them holds some knowledge and tries to maximize it through the social network, where they receive, share, and develop this knowledge (Phelps et al., 2012). Different disciplines, including sociology (Bothner, 2003) and management studies (Reagans & McEvily, 2003), have further studied the influence of the social network on information and knowledge diffusion. Podolny (2001) highlights how the relationships within these networks act as a medium for knowledge sharing and a lens through which they evaluate other agents. Wang (2013) studies how knowledge transfer motivated by knowledge exchange mechanisms in a dynamic set-up affects the evolution of knowledge networks.

Scale-free networks are often used in network algorithms to represent complex real-world networks. The main characteristic of scale-free networks is that their expansion is based on preferential attachment (Barabási & Albert, 1999). Barabási (2009) discusses the universality of scale-free network topology in the several networks present in nature, technology, and societies. For example, it is possible to observe many connections and alliance networks among firms and R&D following a scale-free network pattern (Barabási & Bonobeau, 2003). Barabási-Albert (1999) scale-free network has been widely used across social network analysis as it adopts a "richer gets richer" model, which is not considered by random networks. Liao (2021) explores network mechanisms, mainly preferential attachment of the Barabási network, to study knowledge sharing by users in virtual communities.

3 The Know-It-All Game (KIA)

KIA is a game in which agents share knowledge with their neighbours in a given network. The game is inspired by the PageRank mechanism, the PageRank Game and the Buck-holding game.

The general model of the KIA game consists of a finite number of players represented as vertices of a directed graph. The game begins with Nature evenly allocating knowledge to players which then proceed to share it strategically with one another. In the subsequent iterations of the game, Nature issues new knowledge which is rewarded to players based on the knowledge received by them in the previous iteration. Agents aim to receive knowledge (from neighbours or Nature) as much as possible or, more formally, to maximize personal utility. Utility obtained by players is given by the received knowledge.

Before we move on to the formal definition of the game, we would like to clarify the following definitions:

- Knowledge: In the model, knowledge refers to acquaintance or understanding of scientific and technical information and know-how. Additionally, knowledge is not a product of experience but rather could be understood by anyone with whom it is shared. Knowledge is defined by its complexity and transmissibility which will be discussed further in section 3.2.
- Nature: It refers to the overarching entity/unit that distributes knowledge as per the mechanisms of the game.
- Iterations: The number times this mechanism is repeated to arrive at a stable utility. This is discussed further in sections 3.1 and 3.3.

In the following section, we will provide a formal definition of the KIA game. We will then discuss the parameters, their definitions, and their impact on the game. Finally, we discuss the implicit assumptions and features of our model.

3.1 Formal Definition

Formally, a KIA game Σ is defined in the form $\Sigma = (\mathscr{G}, d, \delta)$. $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is a directed graph in which $\mathscr{V} = \{1, ..., n\}$ is the set of vertices and $\mathscr{E}$ is the set of edges. d represents knowledge complexity and δ represents knowledge transmissibility (sections 3.2.2 and 3.2.3 are entirely dedicated to these two parameters). They are bounded as follows:

$$0 < d < 1\,, \qquad d \in \mathbb{R}\,; \tag{3.1}$$

$$0 < \delta < 1\,, \qquad \delta \in \mathbb{R}\,. \tag{3.2}$$

In the game, $\mathscr{V}$ represents the set of players. For $i \in \mathscr{V}$, call $\mathscr{E}_i$ the set of edges that connects player i with their out-neighbours in $\mathscr{G}$ such that $\mathscr{E}_i = \{(i,j)\,|\,j \in \mathscr{V} : (i,j) \in \mathscr{E}\}$. The set of strategies of player i is $S_i = \mathscr{E}_i$. Note that $(i,i) \in S_i$, as it represents the decision of player i to share their piece of knowledge with no one but themselves. $S = \times_{i \in \mathscr{V}} S_i$ is the set of strategy profiles. We define $\rho_{i,j}(\boldsymbol{x})$ to be the set of edges that describes the shortest network path that connects player i to player j, given a set of edges $\boldsymbol{x}$. For example:

$$\rho_{i,j}(\boldsymbol{x}) = \begin{cases} \{(i,a); (a,b); (b,j)\} & \text{if a path exists} \\ \{\varnothing\} & \text{otherwise} \end{cases} . \tag{3.3}$$

Given a strategy profile $\boldsymbol{s} = \{(s_1); (s_2); ...; (s_n)\}$, where s_i is the strategy selected by player i, the random variable $\Phi_{i,j}(\boldsymbol{s})$ is defined:

$$\Phi_{i,j}(\boldsymbol{s}) = \begin{cases} \delta^{(|\rho_{i,j}(\boldsymbol{s})|-1)} & \text{if } \rho_{i,j}(\boldsymbol{s}) \neq \{\varnothing\} \\ 0 & \text{otherwise} \end{cases} . \tag{3.4}$$

$\Phi_{i,j}$ represents the amount of knowledge possessed by player i which is transmitted to player j directly or indirectly (see section 3.2.3). For example, $\Phi_{i,j} = 1$ means that player j will receive 100% of player i's knowledge, $\Phi_{i,j} = 0$ stands for 0%.

The KIA game has a structure similar to the PageRank algorithm, where payoffs are calculated through an iteration process. Nevertheless, it is important to note that players' strategies do not change over the iterations. The game begins with Nature dividing a unit of knowledge in equal parts and assigning one portion to each player. We name k_i^t the knowledge that player i receives from Nature in iteration t. In the beginning, every agent i receives $k_i^0 = \frac{1}{n}$ knowledge where n is the total number of agents. During any iteration, players simultaneously share the knowledge just received by Nature accordingly to their selected strategy. Note that players share all the knowledge that they possess and knowledge is not "lost" when shared. We name $\bar{k}_i^t$ the total knowledge received by player i during iteration t. $\bar{k}_i^t$ is calculated by:

$$\bar{k}_i^t = k_i^t + \sum_{j=0, j\neq i}^{n} k_j^t \Phi_{j,i} \, . \tag{3.5}$$

At this point, the iteration is concluded.

Every new iteration t begins with Nature issuing a new unit of knowledge to replace the older one. Older knowledge is removed and it will not be counted or shared in any future step. New knowledge is then divided among players proportionally to the level of knowledge achieved in the previous iteration. More precisely:

$$k_i^t = \frac{d \times \bar{k}_i^{t-1}}{\sum_{j=0}^{n} \bar{k}_j^{t-1}} + \frac{1-d}{n} \, . \tag{3.6}$$

Note that this equation is the sum of two elements. $\frac{1-d}{n}$ is identical for every player and stands for the portion of new knowledge that is sporadically created, without the need of a previous finding. Whereas, the first part represents the portion of new knowledge which depends on a previous one. This mechanism rewards players for being knowledgeable and at the same time does not completely exclude less knowledgeable players.

This process of sharing and issuing new knowledge is iterated until iteration t^*, such that $|\bar{k}_i^{t^*-1} - \bar{k}_i^{t^*}| < \epsilon$ for every $i \in \mathscr{V}$, where ϵ is an arbitrarily close to 0. Players receive utility, u_i accordingly to the amount of knowledge received at iteration t^*:

$$u_i = \bar{k}_i^{t^*} \, . \tag{3.7}$$

3.2 Parameters

Every KIA game is shaped by the network structure as well as the parameters d and δ. In this section, we will discuss each parameter with motivations and examples.

3.2.1 The Network

As the players' strategic choices depend on their connection and the position of their connections in the network, the exogenously provided network plays an essential role in the model.

The game is designed such that it can be implemented for any directed graph. Directed graphs represent the connection of nodes through edges in a specified direction. The KIA games should not be bounded by a limited network structure. This way, many different scenarios can be analyzed, from a simple 3-player complete network to much more complex Barabási–Albert networks.

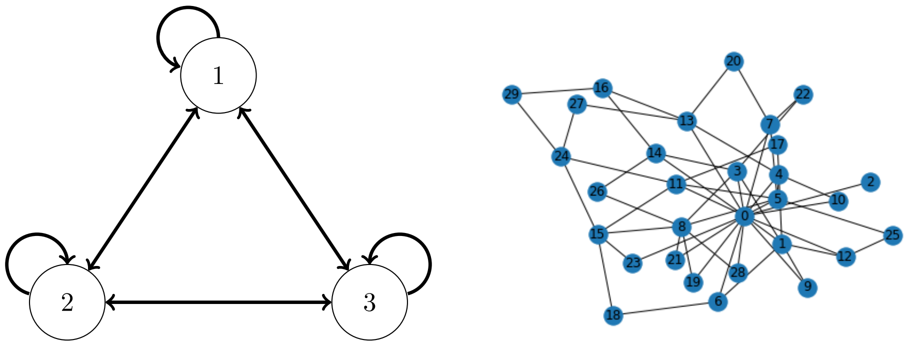


Figure 1: Graphical representation of a 3-player complete network (on the left) and a Barabási–Albert network (on the right). (parameters for the Barabási–Albert network: $n = 30$, $m = 2$)

Complete networks: Complete networks refer to fully connected networks where each node is directly connected to all the nodes. This network structure enables the model to provide an equal positioning to each player at the beginning of the game. Hence, the impact of knowledge transmissibility and knowledge complexity (discussed in the later sections) can be easily observed in the equilibrium strategies.

Barabási networks: The Barabási network allows us to incorporate a scale-free network into the model. As discussed in the literature review, scale-free networks replicate many real-life networks, including academic networks, community networks, the world-wide-web, etc. The most important feature is the asymmetric distribution of connection to the nodes, affecting their relative position in the network. Therefore, this network can provide the model with a more realistic set-up where the players are not equally endowed with similar connections. Testing the model on networks, where certain players are more likely to receive knowledge because of their connections, can provide insight into the real-life patterns of knowledge-sharing.

3.2.2 Complexity of Knowledge: d

Complexity of knowledge is represented by parameter d. It ranges between 1 and 0 where $d = 1 - \epsilon$ stands for complex knowledge and $d = \epsilon$ for basic knowledge.

In a primordial world in which knowledge is extremely basic ($d = \epsilon$), we argue that new discoveries can be made by anyone, regardless of their knowledge background. For example, fire could have been discovered by anyone who went for a walk after a thunderstorm with no need for particular skills. In this world, knowledge distribution at the beginning of every iteration (eq 3.6) would look like:

$$k_i^t = \frac{d \times \bar{k}_i^{t-1}}{\sum_{j=0}^{n} \bar{k}_j^{t-1}} + \frac{1-d}{n} = \frac{1}{n} \,. \tag{3.8}$$

On the other hand, in a highly evolved world in which knowledge is extremely complex ($d = 1 - \epsilon$), we argue that new discoveries can be made only by knowledgeable agents. For example, coding knowledge cannot exist in a society that does not know how to use electricity. In this particular world, knowledge distribution at the beginning of every iteration (eq 3.6) would look like:

$$k_i^t = \frac{d \times \bar{k}_i^{t-1}}{\sum_{j=0}^{n} \bar{k}_j^{t-1}} + \frac{1-d}{n} = \frac{\bar{k}_i^{t-1}}{\sum_{j=0}^{n} \bar{k}_j^{t-1}} \,. \tag{3.9}$$

In this world, agents that do not receive any knowledge are destined not to be relevant from a technological viewpoint as their portion of knowledge becomes smaller and smaller at every iteration. An example of this phenomenon in today's world is the Sentinelese community, an indigenous tribe living on North Sentinel Island, part of the Andaman Islands, an Indian archipelago in the Bay of Bengal. Sentinelese are among the few reclusive societies remaining on Earth, unexposed to the technological state of the world. Returning to our model, it is fairly improbable for the Sentinelese community to develop a piece of knowledge that can contribute to the highly complex knowledge state of the rest of the world, and thus their share of knowledge would be close to zero.

3.2.3 Transmissibility of Knowledge: δ

We believe that knowledge propagates not only through direct connections but also through indirect connections.

Direct knowledge transfer

We define direct knowledge transfer to be an instance in which player i receives k_j directly form player j. Given a strategy profile $\boldsymbol{s}$, a direct knowledge transfer happens whenever:

$$|\rho_{i,j}(\boldsymbol{s})| = 1\ . \tag{3.10}$$

In the following graphical example, player 1 shares knowledge directly with player 2 and vice versa.

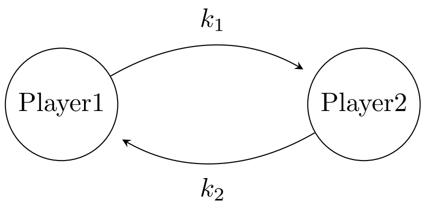


Figure 2: Graphical representation of player 1 and player 2 directly sharing knowledge with each other.

Indirect knowledge transfer

We define indirect knowledge transfer to be an instance in which player i receives k_j not from player j but instead from a player $l \neq j$. Given a strategy profile $\boldsymbol{s}$, an indirect knowledge transfer occurs whenever:

$$|\rho_{i,j}(\boldsymbol{s})| > 1\ . \tag{3.11}$$

Nevertheless, we believe an indirect transfer of knowledge to be less profitable than a direct one. For example: let us consider a scenario in which player 1 shares knowledge with player 2 who, in turn, shares with player 3. In this case, player 2 will learn k_1 directly from player 1 and player 3 will learn k_2 directly from player 2. We argue that player 3 will get to know k_1 indirectly through player 2. We believe that in this indirect transfer of knowledge some value is lost due to “lost in translation” and time-depreciation, which is adjudged through this discount factor.

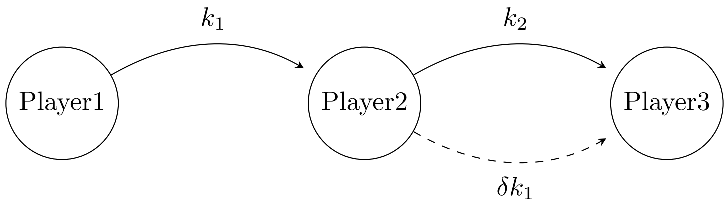


Figure 3: Graphical representation of direct and indirect knowledge transfer.

In order to capture this effect, we implemented the random variable $\Phi_{i,j}$ (eq. 3.4). Every transfer of knowledge is compound discounted by $0 < \delta < 1$ for each player in between the original owner and the receiver of knowledge. Mathematically, this is given by $\delta^{|\rho_{i,j}|-1}$, where $|\rho_{i,j}|$ stands for the number of elements inside the set $\rho_{i,j}$.

3.3 Features of the Model

The KIA game is not only shaped by the parameters themselves but also by their interactions. Features like the creation of new knowledge, iterations, and costs of sharing are peculiar to the model and will be discussed in this section.

3.3.1 Creation of New Knowledge

In every iteration of the game, total new knowledge is normalized to 1. Therefore, what is provided to each player at the beginning of each iteration can be viewed as their share of new knowledge. This can also be regarded as how likely the players are to "create" or "understand" new pieces of knowledge.

No matter how the knowledge is shared in the game, this normalized value of knowledge does not change, only its distribution among the players changes. The total amount of new knowledge, therefore, does not reflect knowledge growth (as it is always equal to 1). Hence, the model is not interested in assessing the growth of knowledge or the creation of new knowledge. It focuses instead on how this "new" knowledge is distributed among the players and how this distribution is impacted by the knowledge-sharing strategies of the player.

3.3.2 Iterations and Utility

In the model, players receive knowledge from Nature depending on the knowledge previously received and the total knowledge shared in the previous period. This follows a similar pattern as the PageRank mechanism, where pages receive scores according to two factors:

- **The number of other pages providing links to the page in question**: The more websites providing links to page A, the higher the score of Page A.
- **The PageRank of these pages**: The higher the PageRank of pages offering links to page A, the higher the score of Page A.

As explained in the literature review section, PageRank solves this problem by calculating the score over many iterations to ascertain a stabilized value. Following a similar approach, the KIA game also looks at the convergence of knowledge received and obtained after many iterations to calculate players' utility. However, differently from PageRank, we believe that iterations in the KIA game have a clear place in reality and are not just artificial constructs. In the model, the players who are receiving many pieces of knowledge are expanding their knowledge share faster than the ones who are occasionally receiving it. In this process, the latter kind of players become obsolete.

3.3.3 Cost of Sharing

In the KIA game, players can share knowledge at no direct cost. However, there are indirect costs. Whenever player 1 shares knowledge with player 2, player 2 becomes more knowledgeable. In turn, this allows player 2 to receive a higher portion of knowledge in the following iteration. And since knowledge is normalized to 1 in each iteration, this results in a smaller portion of knowledge for player 1. Therefore, as a unidirectional transfer, knowledge sharing has an implicit cost. That is, by choosing to share their knowledge with another player, players are indirectly reducing their relative knowledge share in the network.

Intuitively, this should only lead to complete non-sharing of knowledge, however, as we will see in the equilibria and the results section, this is not necessarily the case. Like any other cost-payoff model, despite the cost, the utility received by players can still incentivize them to share their knowledge.

3.3.4 Other Assumptions Regarding the Players

- Players do not lie or share partial knowledge. Whenever a player decides to share their knowledge, all of it is transmitted to the second player and nothing is omitted.
- Every player possesses the intellectual capabilities to understand and bring together knowledge when it is shared with them.
- Players do not communicate or interact in any way other than by sharing knowledge. Therefore, they cannot agree to collude on a strategy that will bring mutual benefits (similar to the prisoner's dilemma case).
- It is important to note that the players are not altruist or revengeful (by design).

4 Simulation Setup

Equilibria in the KIA game can be easily identified when simple and small networks are analysed. However, as the number of players grows, it might become challenging to discover stable patterns and equilibrium points. In order to overcome this issue, we have decided to use a machine learning-based algorithm. Throughout this chapter, we will describe the algorithm, explaining parameters and working dynamics.

4.1 The Algorithm

The algorithm has been coded in Python. It is one of the most accessible programming languages, widely understood, easy and fast to use, and adequate to support the required complexity. The code has been structured in a parametrized manner in order to be scalable and potentially reusable in other Python functions. The algorithm consists of repeated single iterations of a KIA game. The players learn by playing the game as at each iteration they can observe their payoff. Basing decisions on their experience, they select the most profitable strategy (at least according to their forecast). A copy of the code can be found in Appendix B.

4.2 Adaptive Players

In our simulations, players must have the ability to change their minds, adapt their strategies, and try every possibility. Note that this is not the case for the formal model. The KIA game is a one-shot game that uses iterations to calculate payoffs, and thus, strategies are fixed for every iteration. However, only by allowing players to obtain a complete picture of the situation, they can reliably converge on equilibria.

As the KIA game uses iterations to calculate final payoffs, we decided to use machine learning to break into the iteration process. Agents will be allowed to change strategies at every iteration. They build experience by trying strategies and learning from the amount of knowledge received during each iteration, recognizing the differences, improvements, and reductions in payoffs. As players obtain sufficient experience in the initial iterations, they will confidently stick to their selected strategy for the rest of the simulation. The game is then iterated until the final payoffs are reached.

To refer to the strategies selected by each player at every iteration, we add a time dimension to the strategy profile (presented in section 3.1). Therefore, $\boldsymbol{s}^t$ stands for the strategy profile during iteration t and s_i^t stands for the strategy selected by player i during iteration t.

4.3 The Learning Mechanism

The learning mechanism generates the data upon which decisions are taken and strategies are selected. During the game, players observe the received knowledge and build experience. The "Experience matrix", Π, associates an amount of knowledge to each strategy of each player:

$$\Pi = \begin{bmatrix} \pi_{1,1} & \pi_{1,2} & \dots & \pi_{1,j} & \dots & \pi_{1,n} \\ \pi_{2,1} & \pi_{2,2} & \dots & \pi_{2,j} & \dots & \pi_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \pi_{i,1} & \pi_{i,2} & \dots & \pi_{i,j} & \dots & \pi_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \pi_{n,1} & \pi_{n,2} & \dots & \pi_{n,j} & \dots & \pi_{n,n} \end{bmatrix}, \tag{4.1}$$

where $\pi_{i,j}$ is equal to the amount of knowledge that player i believes they will receive as a result of playing strategy (i, j) (share with player j). Moreover, when Π is divided into vectors we obtain single players experiences. More precisely, the row vector $\boldsymbol{\pi}_i$ includes all the beliefs of player i:

$$\boldsymbol{\pi}_i = \begin{bmatrix} \pi_{i,1} & \pi_{i,2} & \dots & \pi_{i,j} & \dots & \pi_{i,n} \end{bmatrix}. \tag{4.2}$$

Note that Π and $\boldsymbol{\pi}_i$ do not represent the real expectations but just the beliefs of the players. These are similar to forecasts and are based on the experience acquired while playing the game. This matrix evolves through iterations and depends on parameters α and o_i^t.

4.3.1 Observed Knowledge Payoff

This subsection is dedicated to the fundamental values assigned to the elements of the "Experience matrix". As they are the cornerstones upon which decisions are taken, they should include direct and indirect payoffs of playing any given strategy.

Let o_i^t be the knowledge payoff observed by player i when playing s_i^t in iteration t. More precisely, we mathematically define o_i^t as:

$$o_i^t = k_i^{t+1} + \sum_{j=0, j \neq i}^{n} k_j^t \Phi_{j,i} \, . \tag{4.3}$$

Note that this is very similar to the total knowledge received in a single iteration $\bar{k}_i^t$ (eq. 3.5). Nevertheless, o_i^t includes k_i^{t+1} instead of k_i^t. In other words, o_i^t counts knowledge received from other players at iteration t plus knowledge received from Nature at iteration $t+1$. This is the case because the amount of knowledge received from Nature during iteration t does not get affected by players' actions during iteration t, however these actions have an effect on knowledge distributed by Nature in iteration $t+1$.

4.3.2 How Experience Evolves: Parameter α

In this subsection, we will clarify how players learn and adapt their Experience vector through iterations.

Alpha, such that $0 \leq \alpha \leq 1$ and $\alpha \in \mathbb{R}$, is a coefficient that represents players' memory. A high α is associated with players that believe the most recent observations to be the most important, forgetting older experiences easily. A low α is associated with players that rely more on past experience, attributing limited importance to new observations.

In formal terms, players update their experience at every iteration depending on the value of α. More precisely:

$$\pi_{i,j}^t = \begin{cases} \left[o_i^t \times \alpha\right] + \left[\pi_{i,j}^{t-1} \times (1-\alpha)\right] & \text{if } s_i^t = (i,j) \\ \pi_{i,j}^{t-1} & \text{otherwise} \end{cases} , \tag{4.4}$$

where $\pi_{i,j}^t$ is the experience value at iteration t that will replace the old value $\pi_{i,j}^{t-1}$.

4.4 The Decision-Making Mechanism

In this final step, the players select their strategies according to their experience. Agents will select the strategy which they believe to be the most profitable. More precisely:

$$s_i^t = (i, j) \quad \text{subject to} \qquad \max \boldsymbol{\pi}_i^t = \pi_{i,j}^t \; . \tag{4.5}$$

In the case in which different values of j satisfy $\max \boldsymbol{\pi}_i^t = \pi_{i,j}^t$, the strategy is determined by randomly selecting one of the equal options.

For example:

Let us consider player 2 experience vector $\boldsymbol{\pi}_2$:

$$\boldsymbol{\pi}_2^t = \begin{bmatrix} 0.3 & 0.5 & 0.3 & 0.8 \end{bmatrix} .$$

In this moment of the game, player 2's experience suggests that sharing knowledge with player 4 will provide 0.8 pieces of knowledge. This value is the highest in the vector, therefore player 2 will set their strategy to be:

$$s_2^t = (2, 4) \; .$$

5 The Complete Network

In this section, we will first analyse the simple 3-player complete network from a theoretical viewpoint. We will proceed by simulating the game on this simple network in order to understand how parameters affect convergence to equilibria and which equilibria occur more frequently. We will then focus on a bigger complete network with 10 players and analyse the game from a theoretical viewpoint and through simulations.

5.1 The 3-Player Complete Network

The 3-Player complete network includes the simplest interactions that we will discuss in our paper. Three players are connected with each other, allowing every agent to choose among three strategies. We will introduce the network with the following graphical representation.

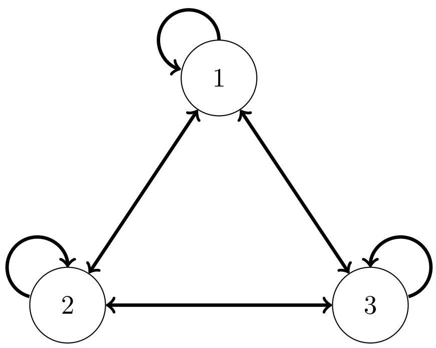


Figure 4: The 3-Player Complete Network

5.1.1 Theoretical Approach

We begin by writing the game in strategic form. However, the payoff calculations depend on many factors and are quite complex. Therefore, we decided to present one set of tables assuming $d = 1$. The complete payoffs are presented in appendix A.2.

$d = 1$**:**

$s_3 = \mathbf{(3,3)}$

s_1 \ s_2	$\mathbf{(2,2)}$	$\mathbf{(2,1)}$	$\mathbf{(2,3)}$
$\mathbf{(1,1)}$	$(\frac{1}{3},\frac{1}{3},\frac{1}{3})$	$(1,0,0)$	$(0,0,1)$
$\mathbf{(1,2)}$	$(0,1,0)$	$(1,1,0)$	$(0,0,1)$
$\mathbf{(1,3)}$	$(0,0,1)$	$(0,0,1)$	$(0,0,1)$

$s_3 = \mathbf{(3,1)}$

s_1 \ s_2	$\mathbf{(2,2)}$	$\mathbf{(2,1)}$	$\mathbf{(2,3)}$
$\mathbf{(1,1)}$	$(1,0,0)$	$(1,0,0)$	$(1,0,0)$
$\mathbf{(1,2)}$	$(0,1,0)$	$(1,1,0)$	$(\frac{2+\delta}{3},\frac{2+\delta}{3},\frac{2+\delta}{3})$
$\mathbf{(1,3)}$	$(1,0,1)$	$(1,0,1)$	$(1,0,1)$

$s_3 = \mathbf{(3,2)}$

s_1 \ s_2	$\mathbf{(2,2)}$	$\mathbf{(2,1)}$	$\mathbf{(2,3)}$
$\mathbf{(1,1)}$	$(0,1,0$	$(1,0,0)$	$(0,1,1)$
$\mathbf{(1,2)}$	$(0,1,0)$	$(1,1,0)$	$(0,1,1)$
$\mathbf{(1,3)}$	$(0,1,0)$	$(\frac{2+\delta}{3},\frac{2+\delta}{3},\frac{2+\delta}{3})$	$(0,1,1)$

Table 1: Strategic representation of the KIA game played on a 3-player complete network ($d = 1$).

We then proceed to find pure Nash equilibria using best-responses. In order not to overload the paper with tables, we present strategy tables with best-responses in Appendix A.2. The tables show the following Nash equilibria:

$$
\begin{aligned}
\mathrm{NE} = \{&(1,1),(2,2),(3,3)\};\{(1,1),(2,1),(3,1)\};\{(1,1),(2,3),(3,2)\};\\
&\{(1,2),(2,1),(3,3)\};\{(1,2),(2,1),(3,1)\};\{(1,2),(2,1),(3,2)\};\\
&\{(1,2),(2,2),(3,2)\};\{(1,2),(2,3),(3,2)\};\{(1,3),(2,3),(3,3)\};\\
&\{(1,3),(2,1),(3,1)\};\{(1,3),(2,2),(3,1)\};\{(1,3),(2,3),(3,1)\};\\
&\{(1,3),(2,3),(3,2)\}\,. \qquad (5.1)
\end{aligned}
$$

However, these equilibria strongly rely on the fact that d is set to 1. In order to understand which of these equilibria also hold true for other values of d, we introduce lemma 1.

Lemma 1:

Any player i who is not receiving any pieces of knowledge from other players is strategically better off by playing strategy (i, i).

This can be proven by recalling the equation 3.5:

$$\bar{k}_i^t = k_i^t + \sum_{j=0, j\neq i}^{n} k_j^t \Phi_{j,i}\,. \qquad (5.2)$$

In a scenario in which player 1 does not receive any pieces of knowledge other than the one received from Nature, we have that $\sum_{j=0,j\neq 1}^{n} k_j^t \Phi_{j,1} = 0$. Therefore by combining this with eq 3.6, we obtain:

$$k_1^t = \bar{k}_1^t = \frac{d \times \bar{k}_1^{t-1}}{\sum_{j=0}^{n} \bar{k}_j^{t-1}} + \frac{1-d}{n}\,. \qquad (5.3)$$

Note that $\sum_{j=0}^{n} \bar{k}_j^{t-1}$ represents the sum of all the knowledge shared in a given period. Therefore, by sharing their knowledge, player 1 increases this value and reduces k_1^t (as $[d \times \bar{k}_1^{t-1}]$ does not increase).

By taking Lemma 1 into account, we can reduce the amount of Nash equilibria presented in eq 5.1 to the following list:

$$\begin{aligned}\text{NE'} = &\{(1,1),(2,2),(3,3)\};\{(1,2),(2,1),(3,3)\};\{(1,1),(2,3),(3,2)\};\\&\{(1,3),(2,2),(3,1)\}\,. \end{aligned} \tag{5.4}$$

These equilibria can be graphically represented as follows:

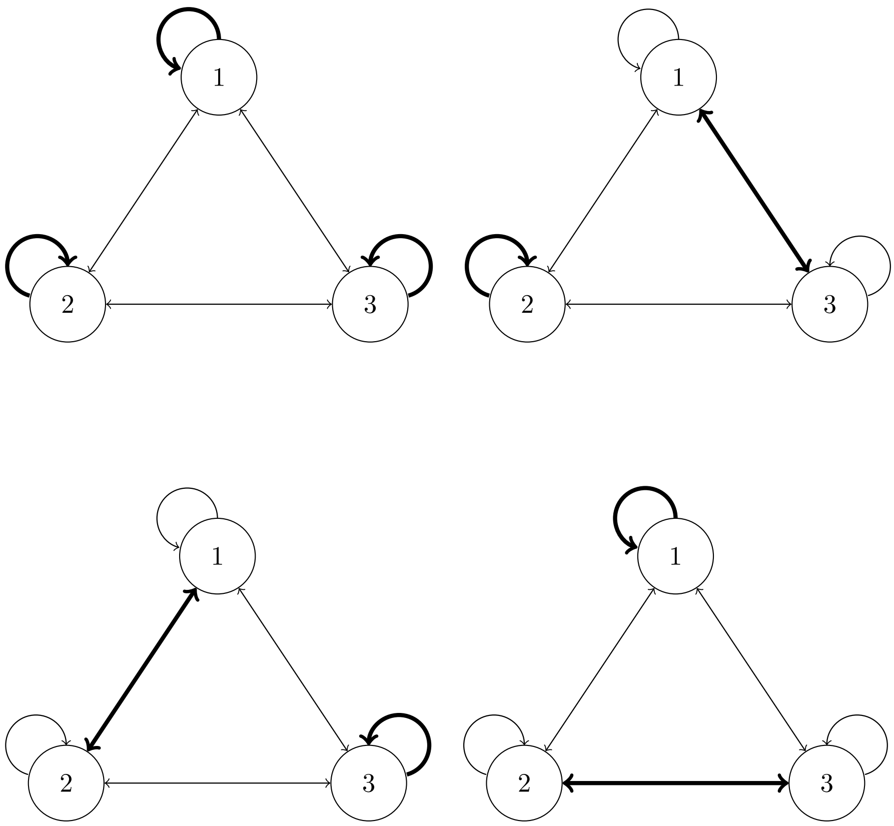


Figure 5: Graphical representation of the equilibria in a 3-player complete network. In bold are the selected strategies. The light arrows represent all the possible non-selected strategies.

For this type of network, there are four pure Nash equilibria. However, they are only of the following 2 kinds:

- **Type 1:** All the players not sharing knowledge with anyone else.
- **Type 2:** Two players sharing knowledge with each other and the third player not sharing knowledge with anyone else.

The first type of equilibrium is clearly proven by Lemma 1 (each agent is better off by not sharing when nobody shares with them). We will now proceed to prove the existence of equilibria for the Type 2 strategies.

Let us consider the following strategy profile $\boldsymbol{s}^* = \{(1,2)(2,1)(3,3)\}$. In this scenario, player 3 is playing their dominant strategy as suggested by Lemma 1. Player 1 can either deviate by playing $(1,1)$ or $(1,3)$. As suggested by table 1, $(1,3)$ is not a viable strategy. In this case, all the knowledge would be channelled to player 3 and player 1 will end up with a lower payoff. Regarding strategy $(1,1)$, it can be said that deviating from $s_1 = (1,2)$ to $s_1 = (1,1)$ given $s_2 = (2,1)$ and $s_3 = (3,3)$ is profitable only if:

$$u_1\{(1,2)(2,1)(3,3)\} - u_1\{(1,1)(2,1)(3,3)\} < 0\,. \tag{5.5}$$

In appendix A.1, the final utility formulas are reported. In order to check inequality 5.5, we will borrow the two needed functions:

$$u_1\{(1,2)(2,1)(3,3)\} = \frac{4d - 1 + \sqrt{16d^2 - 32d + 25}}{6}\,, \tag{5.6}$$

$$u_1\{(1,1)(2,1)(3,3)\} = \frac{4 - d - \sqrt{(d-1)\,(d-4)}}{3}\,. \tag{5.7}$$

The relationship $u_1\{(1,2)(2,1)(3,3)\} - u_1\{(1,1)(2,1)(3,3)\}$ is proven to be positive for every $0 < d < 1$. Therefore, *ceteris paribus* player 1 prefers $s_1 = (1,2)$ to $s_1 = (1,1)$. By symmetry, the same can be inferred about player 2. Again by symmetry, we can conclude that every equilibrium of type 2 is a pure Nash equilibrium.

5.1.2 Machine Learning Approach

Now that we have shown the Nash equilibrium strategies for the 3-player complete network, we can proceed with the simulation. We will observe the results of the algorithm and we will test its accuracy for different values of parameters d and δ as well as learning parameter α.

We run 5999 iterations for each game. As discussed previously, iterations allow the players to converge to a single strategy, stabilising playoffs and utilities. The game is then replayed 100 times. Note that it is not repeated but rather replayed. Players do not have a memory from previous games. We decided to replay the game a sufficient amount of times to limit randomness in results and outliers. In this way, the algorithm provides more consistent results than with longer single games. This number is set to 100 to make the results more understandable and easier to read.

5.1.3 Simulation Results: Parameter d

In this section, we will present the results for the effect of parameter d, knowledge complexity, which is an exogenously provided parameter in the KIA model. Given its crucial role in payoff calculations, we would now like to explore how the parameter's value changes the behaviour of the players in our simulation. We will keep the other parameters constant ($\alpha = 0.15$, $\delta = 0.5$) in order to obtain easily interpretable results (*ceteris paribus*).

The results from the algorithm are as it follows:

d	NE Type 1	NE Type 2	Non-Converging
0.15	2%	95%	3%
0.5	4%	94%	2%
0.85	9%	91%	0%

Table 2: Simulation results: d in the 3-Player complete network. 100 Replays, 5999 Iterations, $\alpha = 0.15$, $\delta = 0.5$

Table 2 summarizes the percentage distribution of Nash Equilibria and non-converging players. The results suggest that a high value of d helps players in finding an equilibrium. This is reasonable as, generally, the higher the d the higher the reward for being knowledgeable (in other words players are incentivized to play competitively). Moreover, the table suggests that the lower the d, the more the equilibria are polarized on Type 2.

We believe that the network size did not allow us to really understand the impact of d on the game. We will repeat the simulation with different values of d and more players in sections 5.2.1, 6.1 and 6.2.

5.1.4 Simulation Results: Parameter δ

In this section, we will present the results for the effect of parameter δ, knowledge transmissibility, which is an exogenously provided parameter in the KIA model. As done with the other parameters, we will keep everything else constant ($\alpha = 0.15$, $d = 0.5$) in order to obtain easily interpretable results (*ceteris paribus*).

The results from the algorithm are as it follows:

δ	NE Type 1	NE Type 2	Non-Converging
0.2	0%	100%	0%
0.5	4%	94%	2%
0.8	11%	82%	7%

Table 3: Simulation results: δ in the 3-Player complete network. 100 Replays, 5999 Iterations, $\alpha = 0.15$, $d = 0.5$

Table 3 summarizes the percentage distribution of Nash Equilibria and non-converging players. The results suggest that a low value of δ highly polarizes the game towards equilibria of Type 2. This is reasonable as a high δ rewards the creation of loops of many players. However, these results show that a high δ allows different types of equilibria to be reached at the cost of diminished accuracy.

Again, the network size did not allow us to really understand the impact of δ on the game. We will repeat the simulation with different values of δ and more players in the section 5.2.2.

5.1.5 Selection of Parameter α

To recap from previous sections, the learning mechanism of the players depends on the parameter α. A higher α represents a higher trust in the most recent observation, and a lower α represents a higher trust in the experience collected during the game.

Keeping this variation in mind, we identify 3 values that we will use to test the model:

- $\alpha = 0.05$,
- $\alpha = 0.15$,
- $\alpha = 0.5$.

We test these three values for $d = 0.5$ and $\delta = 0.5$. The values of these parameters are kept at their mid-point to simplify our analysis (*ceteris paribus*).

The results from the algorithm are as it follows:

α	NE Type 1	NE Type 2	Non-Converging
0.5	4%	91%	5%
0.15	4%	94%	2%
0.05	17%	83%	0%

Table 4: Simulation results: α in the 3-Player complete network. 100 Replays, 5999 Iterations, $d = 0.5$, $\delta = 0.5$

Table 4 summarizes the percentage distribution of Nash Equilibria and non-converging players. It can be interpreted as follows:

- $\alpha = 0.5$ - Players converge to one of the two equilibria 95% of the times. This set up seems to favour Type 2 equilibria. However, 5 % of the games do not converge to any Nash equilibria.
- $\alpha = 0.15$ - Players converge to one of the two equilibria 98% of the times, strongly favouring Type 2 equilibria.
- $\alpha = 0.05$ - Players converge to one of the two equilibria 100% of the times. In this setup, Type 1 equilibria are met a substantial amount of times.

Based on these results, we decided to use $\alpha = 0.05$ in future simulations. Results showed 0% non-converging players as well as a good variety of equilibria. Note that there are 3 different Type 2 equilibria while only one Type 1. This might be one of the reasons behind the high percentages of Type 2 equilibria in the above results.

5.2 The 10-Player Complete Network

This section aims to scale up the model for more extensive networks through machine learning simulation. Given that the players are symmetrically connected, the scope for knowledge-sharing also significantly increases. To observe the converging knowledge-sharing patterns and loops over the changing parameters, we will run simulations for different values of d and δ. The 10-Player complete network is illustrated in the following figure.

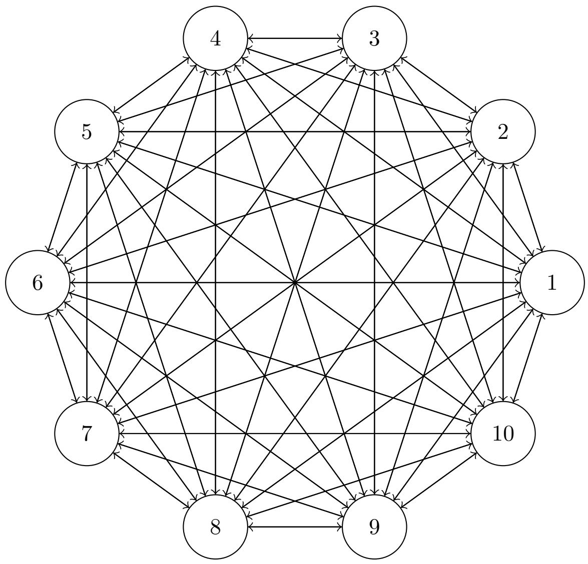


Figure 6: The 10-Player Complete Network

The simulation is run for 10 players in a complete network. The players play the game for 5999 iterations. In the previous section, $\alpha = 0.05$ was shown to offer the most consistent converging patterns. Therefore, α will be set to 0.05 for the whole simulation. The numbers related to the knowledge-sharing loops are calculated over 100 replays of the game.

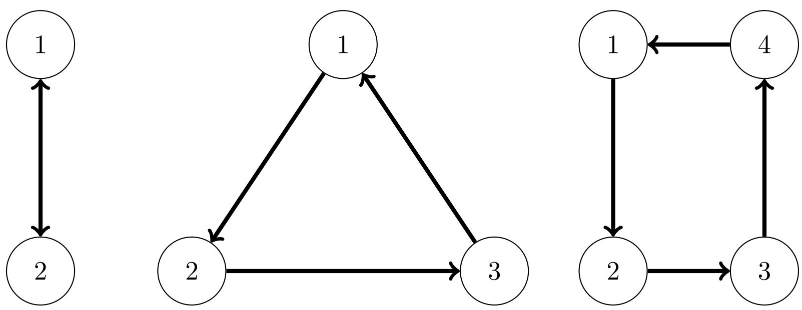


Figure 7: Example of knowledge-sharing loops. From left to right: 2-player loop, 3-player loop, and 4-player loop

5.2.1 Simulation Results: Parameter d

We will run three set of simulations, one for each value of d that we are testing for. The values are: $d = 0.85$, $d = 0.5$ and $d = 0.15$. All the other parameters will not vary ($\delta = 0.5$ and $\alpha = 0.05$).

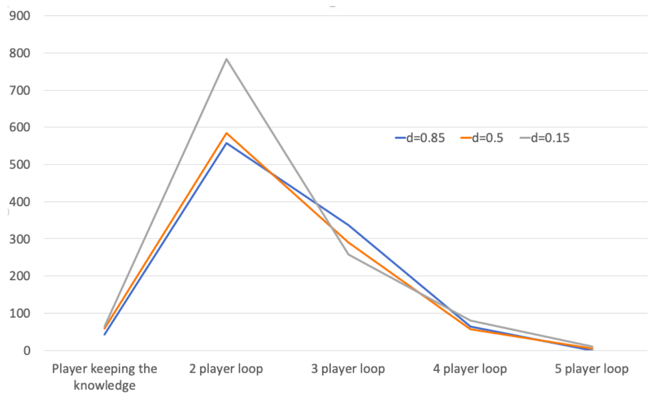


Figure 8: Simulation results: d in the 10-player complete network. 100 Replays, 5999 Iterations, $\alpha = 0.05$, $\delta = 0.5$. x-axis: Dimension of sharing-loops. y-axis: Players involved in a x dimensional sharing-loop

d	0.15	0.5	0.85
Non-Conv. Players	0.4%	0.5%	0.2%

Table 5: Non-Converging Players in the 10-player complete network. Analysis: d

Considering that the network size increased, setting $\alpha = 0.05$ proved to be a sound decision. The percentages of non-converging players are low and similarly distributed across d.

We observe that, in large networks like this, there is a possibility of players converging to stable knowledge-sharing loops that are bigger in size. It is interesting to see that while a 3-player loop is not a stable equilibrium for a 3-player game, as the players will always be better off by deviating and not sharing the knowledge (shown in section 5.1.1), this is not the case for 10 players.

Furthermore, as shown in Figure 8, the average sharing-loop size is increasing through d. Intuitively, it can be comprehended as follows: for higher knowledge complexity, the player's knowledge portion depends heavily on the received knowledge from neighbours (the share of new knowledge distributed randomly by nature is significantly small). Therefore, the incentive to receive more and more knowledge from neighbours is also higher, resulting in players converging into bigger knowledge-sharing loops.

5.2.2 Simulation Results: Parameter δ

We will run three set of simulations, one for each value of δ we are testing for. The values are: $\delta = 0.8$, $\delta = 0.5$ and $\delta = 0.2$. All the other parameters will not vary ($d = 0.5$ and $\alpha = 0.05$).

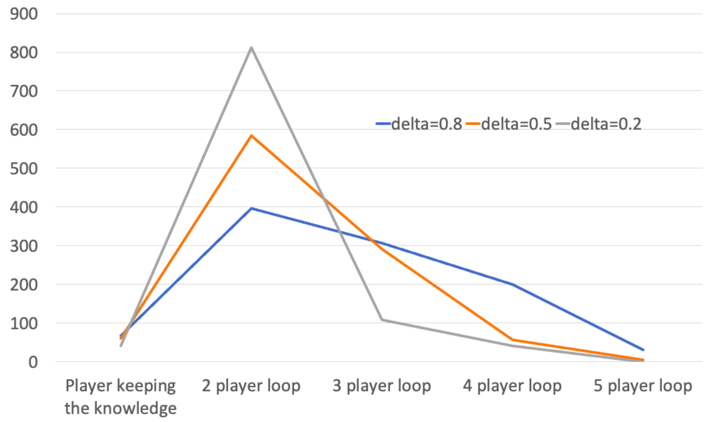


Figure 9: Simulation results: δ in the 10-player complete network. 100 Replays, 5999 Iterations, $\alpha = 0.05$, $d = 0.5$. x-axis: Dimension of sharing-loops. y-axis: Players involved in a x dimensional sharing-loop

δ	0.2	0.5	0.8
Non-Conv. Players	0%	0.5%	0.1%

Table 6: Non-Converging Players in the 10-player complete network. Analysis: δ

As in the previous section (5.2.1), the percentages of non-converging players are low.

Intuitively, data suggests that the higher knowledge transmissibility the bigger the variety of sharing loops. When $\delta = 0.8$ (blue line) similar amounts of players get involved in 2, 3, and 4-player sharing loops. This can be explained by the fact that when knowledge is highly transmissible, players are highly rewarded for being involved in big loops.

However, as the value of δ decreases, players tend to polarize into pairs, sharing reciprocally. As suggested during the 3-player complete network simulations, decreasing knowledge transmissibility removes incentives for big loops. Moreover, smaller loops are generally easier to form. These two factors combined almost guarantee players to end up in a 2-player loop.

6 The Barabási–Albert Network

We will continue our simulation journey by moving to real-life networks. In this section, we aim to extend the model's applications to more complex networks. Backed by the theoretical understanding discussed in the previous sections, we will play the Know-It-All game on networks created using the Barabási–Albert model.

As discussed in previous chapters, we conduct simulations using scale-free Barabási networks. A directed graph created using the Barabási–Albert model is provided to the algorithm by specifying n, the number of nodes (players), and m, the number of edges (links) established between the nodes. This results in an uneven distribution of connections among the nodes, with some nodes becoming more central and well-connected.

Our focus will be on observing the nature of knowledge-sharing loops and how they vary over the parameter d for two different kinds of Barabási networks of 30 players. First, we will set $m = 1$. This network contains many players with a single neighbour and sharing loops of more than 2 players are impossible (because of the structure of the network). Then we will set $m = 3$. In this case, players will be well connected with each other, allowing bigger sharing loops to form.

In this section, we will run simulations varying only networks and knowledge complexity (d). Learning parameter α will be set at 0.05. Knowledge transmissibility (δ) will be set at 0.8. We decided to keep δ constant in order to simplify the analysis and results. We decided to set $\delta = 0.8$ because it allows for different loop sizes (shown in the previous section) and does not polarize the results.

6.1 Results from Network 1: $n = 30$ and $m = 1$

This network consists of 30 players, with at least one connection each (example in Figure 10). In our simulation, agents play the game for 9999 iterations. As previously explained, we set $\alpha = 0.05$ and $\delta = 0.8$. Then the game is replayed 100 times. Note that on each new game, the network is replaced with a new randomly generated graph with the same characteristics.

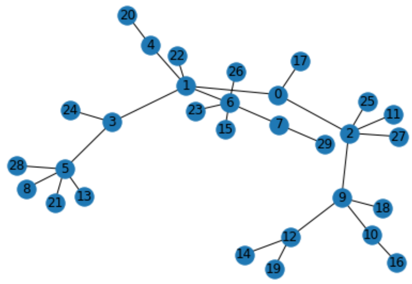


Figure 10: Example of a network obtained using the Barabási–Albert model. $n = 30$ and $m = 1$.

We run the model for the same 3 values of d used in the previous sections: $d = 0.85$, $d = 0.5$ and $d = 0.15$. The results are presented in the following table:

d	Not Sharing	Sharing	Non-Converging
0.15	32.2%	50.5%	17.3%
0.5	30.7%	52.1%	17.2%
0.85	25.5%	59.6%	14.9%

Table 7: Simulation results: d in the Barabási–Albert network. 100 Replays, 9999 Iterations, $\alpha = 0.05$, $\delta = 0.8$, $n = 30$, $m = 1$

It appears clear that the structure of this network rarely allowed players to try different strategies. Agents are often connected to only one other player, who is in turn connected to many other players. In this situation, the "famous" neighbour will share knowledge back only with one of their connections, making many players solitary. Here, the "famous" player refers to the player with more connections, like Player 5 in Figure 10. This phenomenon explains why we see many players not sharing knowledge with anyone else.

Nevertheless, there is a visible pattern in results: sharing slightly grows as the knowledge complexity d increases. At the same time, the amount of non-converging players decreases with the value of d.

To conclude, we have to acknowledge that the ratio of non-converging players in this set of simulations was extremely high. Thus, the results are likely inaccurate. The cause might be, as mentioned before, the particular structure of the network, not allowing many interactions to take place.

6.2 Results from Network 2: $n = 30$ and $m = 3$

In this section, we will simulate the KIA game on a Barabási–Albert model network with parameters $n = 30$ and $m = 3$. The other parameters are set as in the previous section.

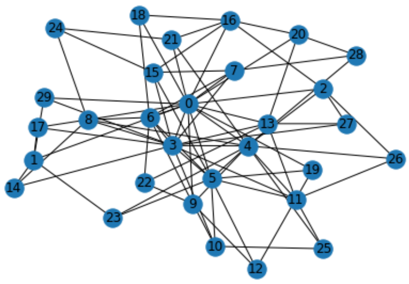


Figure 11: Example of a network obtained using the Barabási–Albert model. $n = 30$ and $m = 3$.

Again, we run the model for 3 values of d: $d = 0.85$, $d = 0.5$ and $d = 0.15$. The following graph and table represent the results.

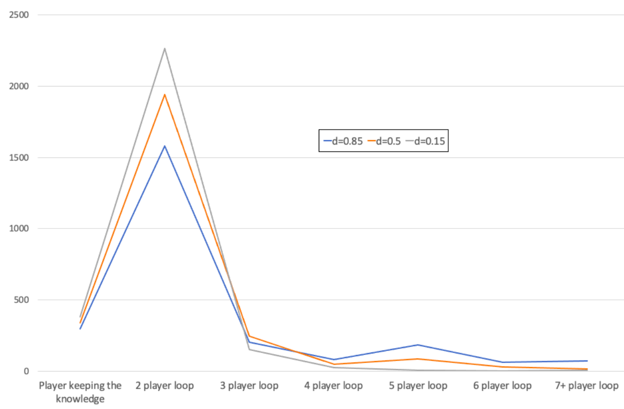


Figure 12: Simulation results: d in the Barabási–Albert network. 100 Replays, 9999 Iterations, $\alpha = 0.05$, $\delta = 0.8$, $n = 30$, $m = 3$. x-axis: Dimension of sharing-loops. y-axis: Players involved in a x dimensional sharing-loop

d	0.15	0.5	0.85
Non-Conv. Players	3.1%	4.1%	6.4%

Table 8: Simulation results: d in the Barabási–Albert network. 100 Replays, 9999 Iterations, $\alpha = 0.05$, $\delta = 0.8$, $n = 30$, $m = 3$

These results are more convincing than the ones obtained in the previous section. In these simulations, the non-converging players were around 5% for every value of d. Even if this is still a high ratio, considering the dimensions and the complexity of the network, we can consider the simulation satisfactory. The results suggest:

- Knowledge complexity (d) increases dispersion in the size of sharing loops, thus increasing the average size of loops. This result can be interpreted as follows: players cooperate better to form bigger loops when knowledge is complex. In comparison with the other distributions, the fat tail of the $d = 0.85$ distribution suggests that, after a certain threshold, players might become substantially more willing to share.
- In this network, players rarely isolate themselves and often end up sharing knowledge with each other. This aspect is weakly affected by knowledge complexity (d), as the graph shows a similar amount of single players across d.
- The two simulated Barabási–Albert networks ($m = 1$ and $m = 3$) share a common trend. In both cases, simulation accuracy increases in d.
- 2-player sharing loops are the most common outcome for every value of d. This can be explained by the fact that it is easier for two players to converge on a strategy than it would be for three players.

7 Discussions

In this thesis, we developed a model that aims to interpret how knowledge is distributed in a network. The purpose of the model is, therefore, to simplify and provide a set-up that fits with the real-life behaviour of agents. However, like any model which aims to boil down the complexities of the real world to mathematical constructs, this model faces some limitations. Some of these limitations offer possibilities for further model development, while others require a different approach. In this section, we would like to bring these discussions into light, further improving the scope and applicability of the model.

Firstly, this paper mainly focuses on a deterministic model. More understandably, players are only allowed to play pure strategies (e.g., Player 1 can choose player 2 as his strategy with a probability of 1 or 0 for every iteration). While this provides the model with some interesting insights about the player's knowledge-sharing behaviour and equilibria, it leaves out the mixed-nash equilibria and mixed strategies. By adapting the model to a stochastic game, the potential related to the observed player's strategies, equilibria, and knowledge-sharing loops can be expanded further.

Secondly, the model does not account for knowledge growth caused by the spread of knowledge through the network. It can be argued that a society in which knowledge spreads widely should grow quicker than a society of egoists. Our model only accounts for how its relative values are provided to the players, not considering overall growth. This setup works when players are more interested in their personal share than the overall growth in knowledge. However, accounting for side-utility obtained by players because of overall improvements in knowledge levels can be quite complicated. One way to incorporate this is to make the parameter d dynamic within the game.

Thirdly, we assume that players interact on a static network. This allows us to observe the players' interactions with lesser variations in the background. Nevertheless, real-life networks are often more dynamic and evolve as per the interactions of their nodes. Allowing the network to evolve through iterations is an interesting path, making the model more adaptable to real-life networks.

Fourthly, in a fast globalizing world, we can observe that knowledge can be shared with more than one player at the time. The KIA game only allows players to select one player as their knowledge-sharing strategy. Under specific premises, the assumption can hold up. For example, to be in a knowledge-sharing relationship with one other agent in a network of 20 players makes more sense than it would for a network of 1000 players. Additionally, only certain kinds of knowledge can be transferred in a closed network. However, going forward with a higher number of players and with a specific definition of knowledge would require more inquiry into this assumption of the model.

Finally, we believe that the KIA game provides us with a good fundamental understanding of the knowledge-sharing theory and patterns, which we would now like to test on some real-life data like research networks and historical knowledge-sharing networks.

8 Conclusions

The idea's genesis for this thesis started with a simple question: Why are some agents more knowledgeable than others?

Through this paper, we have attempted to come closer to the answer to this question. The KIA game shows that agents can share knowledge among themselves and arrive at stable equilibria. However, in the KIA game, no player can obtain the most utility alone. Cooperation is the only way to "win" the game. For example: in a world where nobody shares knowledge, 2 players can take the market simply by cooperating. In turn, 3 other players may join the forces to dethrone the other 2. This chain can repeat many times. However, when the cooperation circle becomes big enough, players may find it profitable to betray their companions, incentivizing everyone to leave the coalition (returning to the world where nobody shares).

Additionally, we observed that the game often stabilizes on scenarios where certain players are better off than others. This indicates that the exclusion of a few players from the knowledge loops could be out of design. After all, monopolizing the knowledge share among the few players leads to higher relative knowledge for them, and an equal distribution benefits none. This is an important insight to understand the distribution of knowledge and the network patterns that we can observe worldwide.

While the model has essentially focussed on understanding the knowledge-sharing and distribution processes in a network, other applications can be explored. For instance, the model can be used within welfare and care context. Let us consider a model similar to the KIA game in which the value shared by the player is care instead of knowledge, which then translates to welfare for the receiving player. We can view this as an iterative process where higher care leads to higher welfare, which in turn provides players with a higher amount of care to share. By finding the right setting for every parameter, we can observe the size, nature, and distribution of welfare among the care-units formed (in terms of family, partners, or communities) in society. However, as people usually provide care to multiple individuals with varying degrees, a stochastic version of the game is needed to exploit the model's full potential.

We had many discussions and considerations regarding the implementation of some mechanisms of "punishment" or "war" while developing the model. We agreed that the model's best feature was not to have harmful threats. With this in place, we played the game on many different networks with a dozen parameters and obtained numerous patterns and equilibria. However, the most recurring social structure was made by two players reciprocally sharing with one another. This is a curious fact, as (in the algorithm) players had no way of knowing who shared knowledge with them.

9 Glossary

Glossary arranged by sections. The terms are located in the section in which they first appeared and in order of appearance.

Section 3:

- KIA game: Know-It-All game.

Section 3.1:

- Σ: A Know-It-All game.
- $\mathscr{G}(\mathscr{V}, \mathscr{E})$: A directed graph.
- d: Parameter that represents knowledge complexity.
- δ: Parameter that represents knowledge transmissibility.
- $\mathscr{V}$: The set of vertices (players).
- $\mathscr{E}$: The set of edges (connections).
- n: The number of players. (also parameter of Barabási-Albert model)
- $\mathscr{E}_i$: The set of edges that connect player i to their out neighbours.
- (i, j): Edge that connects player i to player j in a network.
- S_i: The set of strategies of Player i.
- S: The set of strategy profiles.
- $\rho_{i,j}(\boldsymbol{x})$: The set of edges that describes the shortest network path that connects player i to player j in the set of edges $\boldsymbol{x}$.
- $\{\varnothing\}$: The empty set.
- $\boldsymbol{s}$: A strategy profile in a Know-It-All game.
- s_i: The strategy selected by player i.
- $|\rho_{i,j}|$: The number of elements in set $\rho_{i,j}$.
- $\Phi_{i,j}(\boldsymbol{s})$: The amount of knowledge possessed by player i which is transmitted to player j directly or indirectly.
- k_i^t: The knowledge that player i receives from Nature in iteration t
- $\bar{k}_i^t$: The total knowledge received by player i during iteration t.
- u_i: Utility of player i.
- t: An iteration

Section 3.2

- m: Parameter used in Barabási-Albert Model Networks.

Section 4.2

- $\boldsymbol{s}^t$: In simulation, the strategy profile during iteration t.
- s_i^t: In simulation, the strategy selected by player i during iteration t.

Section 4.3

- Π: The Experience Matrix.
- $\pi_{i,j}$: In simulation, the amount of knowledge that player i believes they will receive as a result of playing strategy (i, j).
- $\boldsymbol{\pi}_i$: In simulation, the vector that includes all the believes of player i regarding each strategy
- o_i^t: In simulation, the knowledge payoff observed by player i when playing s_i^t in iteration t.
- α: In simulation, the learning parameter represents players' memory.

Section 5.1

- NE: Nash Equilibrium / Nash Equilibria.

10 Bibliography

Anand, A., Muskat, B., Creed, A., Zutshi, A. and Csepregi, A. (2021) 'Knowledge sharing, knowledge transfer and SMEs: evolution, antecedents, outcomes and directions', *Personnel review*, 50(9), pp.1873-1893.

Al-Gharaibeh, R.S. and Ali, M.Z. (2022) 'Knowledge sharing framework: A game-theoretic approach', *Journal of the Knowledge Economy*, 13(1), pp.332-366.

Antonelli, C. (2006) 'The business governance of localized knowledge: an information economics approach for the economics of knowledge', *Industry and Innovation*, 13(3), pp.227-261.

Arrow, K. (1962) 'Economic welfare and the allocation of resources for invention', in Universities-National Bureau Committee for Economic Research, (ed.) *The Rate and Direction of Inventive Activity: Economic and Social Factors*. Princeton University Press, pp.609-626.

Atkinson, A.B. and Stiglitz, J.E. (1969) 'A new view of technological change', *The Economic Journal*, 79(315), pp.573-578.

Barabási, A.L. and Albert, R. (1999) 'Emergence of scaling in random networks', *science*, 286(5439), pp.509-512.

Barabási, A.L. and Bonabeau, E. (2003) 'Scale-free networks', *Scientific american*, 288(5), pp.60-69.

Barabási, A.L. (2009) 'Scale-free networks: a decade and beyond', *Science*, 325(5939), pp.412-413.

Borgatti, S.P. and Halgin, D.S. (2011) 'On network theory', *Organization science*, 22(5), pp.1168-1181.

Bothner, M.S. (2003) 'Competition and social influence: The diffusion of the sixth-generation processor in the global computer industry', *American Journal of Sociology*, 108(6), pp.1175-1210.

Brass, D.J. (2002) 'Social networks in organizations: Antecedents and consequences', Unpublished.

Brin, S. and Page, L. (1998) 'The anatomy of a large-scale hypertextual Web search engine', *Computer Networks and ISDN Systems*, 30(1), pp.107–117.

Cabrera, A. and Cabrera, E.F. (2002) 'Knowledge-sharing dilemmas' *Organization studies*, 23(5), pp.687-710.

Chaudhary, N., Salali, G.D., Thompson, J., Rey, A., Gerbault, P., Stevenson, E.G.J., Dyble, M., Page, A., Smith, D., Mace, R. and Vinicius, L. (2016) 'Competition for Cooperation: variability, benefits and heritability of relational wealth in hunter-gatherers', *Scientific Reports*, 6(1), pp.1-7.

Cominetti, R., Quattropani, M. and Scarsini, M. (2022) 'The buck-passing game', *Mathematics of Operations Research*, 47(3), pp.1731-1756.

Conley, T.G. and Udry, C.R. (2010) 'Learning about a new technology: Pineapple in Ghana', *American economic review*, 100(1), pp.35-69.

Collins, C.J. and Smith, K.G. (2006) 'Knowledge exchange and combination: The role of human resource practices in the performance of high-technology firms', *Academy of management journal*, 49(3), pp.544-560.

Cummings, J. (2003) 'Knowledge sharing: A review of the literature', *Washington, DC: World Bank.*

Cummings, J.N. (2004) 'Work groups, structural diversity, and knowledge sharing in a global organization', *Management science*, 50(3), pp.352-364.

Davenport, T.H. and Prusak, L. (1998) *Working knowledge: How organizations manage what they know.* Harvard Business Press.

Fehr, E. and Gächter, S. (2000) 'Fairness and retaliation: The economics of reciprocity', *Journal of economic perspectives*, 14(3), pp.159-181.

Feldstein, M., Horioka, C. and Savings, D. (1992) 'The Solow growth model', *Quarterly Journal of Economics*, 107(2), pp.407-437.

Foray, D. (2004) *Economics of knowledge.* MIT press.

Galunic, D.C. and Rodan, S. (1998) 'Resource recombinations in the firm: Knowledge structures and the potential for Schumpeterian innovation', *Strategic management journal*, 19(12), pp.1193-1201.

Grossman, G.M. and Helpman, E. (1994) 'Endogenous innovation in the theory of growth', *Journal of Economic Perspectives*, 8(1), pp.23-44.

Guedon, J.C. (1977) 'Michel Foucault: the knowledge of power and the power of knowledge', *Bulletin of the History of Medicine*, 51(2), pp.245-277.

Hansen, M.T. (1999) 'The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits', *Administrative science quarterly*, 44(1), pp.82-111.

Hayek, F.A. (1945) 'The use of knowledge in a society', *The American Economic Review*, 35(4), pp.519-530.

Haythornthwaite, C. (1996) 'Social network analysis: An approach and technique for the study of information exchange', *Library & information science research*, 18(4), pp.323-342.

Hopcroft, J. and Sheldon, D. (2008) *Network reputation games*. Cornell University.

Hung, S.Y., Durcikova, A., Lai, H.M. and Lin, W.M. (2011) 'The influence of intrinsic and extrinsic motivation on individuals' knowledge sharing behaviour', *International Journal of Human-Computer Studies*, 69(6), pp.415-427.

Kelly, M. and O Grada, C. (2000) 'Market Contagion: Evidence from the Panics of 1854 and 1857', *American Economic Review*, 90(5), pp.1110-1124.

Kimmerle, J., Moskaliuk, J. and Cress, U. (2011) 'Using wikis for learning and knowledge building: Results of an experimental study', *Journal of Educational Technology & Society*, 14(4), pp.138-148.

Lam, A. and Lambermont-Ford, J.P. (2010) 'Knowledge sharing in organisational contexts: a motivation-based perspective',*Journal of knowledge management*, 14(1), pp.51-66.

Liao, C.H. (2021) 'Exploring the impacts of network mechanisms on knowledge sharing and extra-role behaviour', *Journal of Knowledge Management*, 26(8), pp.1901-1920.

Lin, H.F. (2007) 'Knowledge sharing and firm innovation capability: an empirical study', *International Journal of manpower*, 28(3/4), pp.315-332.

Machlup, F. (2014) *Knowledge: its creation, distribution and economic significance*. Princeton university press. Volume I.

Mokyr, J. (2005) 'Long-term economic growth and the history of technology' *In Handbook of economic growth*, 1, pp.1113-1180.

Ohlsson, S. (2011) *Deep learning: How the mind overrides experience*. Cambridge University Press.

Paier, M. and Scherngell, T. (2011) 'Determinants of collaboration in European R&D networks: empirical evidence from a discrete choice model', *Industry and innovation*, 18(1), pp.89-104.

Pasquinelli, M. (2009) 'Google's PageRank algorithm: A diagram of cognitive capitalism and the rentier of the common intellect', *Deep search: The politics of search beyond Google*, pp.152-162.

Perez-Trujillo, M. and Lacalle-Calderon, M. (2020) 'The impact of knowledge diffusion on economic growth across countries', *World Development*, 132, p.104995.

Phelps, C., Heidl, R. and Wadhwa, A. (2012) 'Knowledge, networks, and knowledge networks: A review and research agenda', *Journal of management*, 38(4), pp.1115-1166.

Podolny, J.M., (2001) 'Networks as the pipes and prisms of the market', *American journal of sociology*, 107(1), pp.33-60.

Reagans, R. and McEvily, B. (2003) 'Network structure and knowledge transfer: The effects of cohesion and range', *Administrative science quarterly*, 48(2), pp.240-267.

Romer, P.M. (1986) 'Increasing returns and long-run growth', *Journal of political economy*, 94(5), pp.1002-1037.

Romer, P. (1993) 'Idea gaps and object gaps in economic development', *Journal of monetary economics*, 32(3), pp.543-573.

Saint-Paul, G (2003), 'Information Sharing and Cumulative Innovation in Business Networks', *CEPR Press*, Discussion Paper No. 4116.

Stauffer, D. (1999) 'Why People Hoard Knowledge: Employees' tendency to jealously guard what they know represents a threat to your business', *Across the board*, 36, pp.16-20.

Strulik, H. (2014) 'Knowledge and growth in the very long run', *International Economic Review*, 55(2), pp.459-482.

Wang, X. (2013) 'Forming mechanisms and structures of a knowledge transfer network: theoretical and simulation research', *Journal of Knowledge Management*, 17(2), pp.278-289.

Wasko, M.M. and Faraj, S. (2005) 'Why should I share? Examining social capital and knowledge contribution in electronic networks of practice', *MIS quarterly*, pp.35-57.

Yang, H. L., and Wu, T. C. (2008) 'Knowledge sharing in an organization', *Technological Forecasting and Social Change*, 75(8), pp.1128-1156.

A Appendix 1: The 3-Player Complete Network

A.1 The Best Response Table

$d = 1$:

$s_3 = \mathbf{(3,3)}$ s_2

		$\mathbf{(2,2)}$	$\mathbf{(2,1)}$	$\mathbf{(2,3)}$
	$\mathbf{(1,1)}$	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	$(1, 0, 0)$	$(0, 0, 1)$
s_1	$\mathbf{(1,2)}$	$(0, 1, 0)$	$(1, 1, 0)$	$(0, 0, 1)$
	$\mathbf{(1,3)}$	$(0, 0, 1)$	$(0, 0, 1)$	$(0, 0, 1)$

$s_3 = \mathbf{(3,1)}$ s_2

		$\mathbf{(2,2)}$	$\mathbf{(2,1)}$	$\mathbf{(2,3)}$
	$\mathbf{(1,1)}$	$(1, 0, 0)$	$(1, 0, 0)$	$(1, 0, 0)$
s_1	$\mathbf{(1,2)}$	$(0, 1, 0)$	$(1, 1, 0)$	$(\frac{2+\delta}{3}, \frac{2+\delta}{3}, \frac{2+\delta}{3})$
	$\mathbf{(1,3)}$	$(1, 0, 1)$	$(1, 0, 1)$	$(1, 0, 1)$

$s_3 = \mathbf{(3,2)}$ s_2

		$\mathbf{(2,2)}$	$\mathbf{(2,1)}$	$\mathbf{(2,3)}$
	$\mathbf{(1,1)}$	$(0, 1, 0$	$(1, 0, 0)$	$(0, 1, 1)$
s_1	$\mathbf{(1,2)}$	$(0, 1, 0)$	$(1, 1, 0)$	$(0, 1, 1)$
	$\mathbf{(1,3)}$	$(0, 1, 0)$	$(\frac{2+\delta}{3}, \frac{2+\delta}{3}, \frac{2+\delta}{3})$	$(0, 1, 1)$

Table 9: Strategic representation of the KIA game played on a 3-player complete network. $d = 1$.

A.2 Players' Utility Functions

$$u_1\{(1,1)(2,2)(3,3)\} = \frac{1}{3} \tag{A.1}$$

$$u_2\{(1,1)(2,2)(3,3)\} = \frac{1}{3} \tag{A.2}$$

$$u_3\{(1,1)(2,2)(3,3)\} = \frac{1}{3} \tag{A.3}$$

$$u_1\{(1,2)(2,3)(3,1)\} = \frac{2+\delta}{3} \tag{A.4}$$

$$u_2\{(1,2)(2,3)(3,1)\} = \frac{2+\delta}{3} \tag{A.5}$$

$$u_3\{(1,2)(2,3)(3,1)\} = \frac{2+\delta}{3} \tag{A.6}$$

$$u_1\{(1,3)(2,1)(3,2)\} = \frac{2+\delta}{3} \tag{A.7}$$

$$u_2\{(1,3)(2,1)(3,2)\} = \frac{2+\delta}{3} \tag{A.8}$$

$$u_3\{(1,3)(2,1)(3,2)\} = \frac{2+\delta}{3} \tag{A.9}$$

$$u_1\{(1,2)(2,1)(3,3)\} = \frac{4d-1+\sqrt{16d^2-32d+25}}{6}, \tag{A.10}$$

$$u_2\{(1,2)(2,1)(3,3)\} = \frac{4d-1+\sqrt{16d^2-32d+25}}{6} \tag{A.11}$$

$$u_3\{(1,2)(2,1)(3,3)\} = \frac{7-4d-\sqrt{16d^2-32d+25}}{6} \tag{A.12}$$

$$u_1\{(1,1)(2,1)(3,3)\} = \frac{4-d-\sqrt{(d-1)(d-4)}}{3}. \tag{A.13}$$

$$u_2\{(1,1)(2,1)(3,3)\} = \frac{d-1+\sqrt{(d-1)(d-4)}}{3} \tag{A.14}$$

$$u_3\{(1,1)(2,1)(3,3)\} = \frac{d-1+\sqrt{(d-1)(d-4)}}{3} \tag{A.15}$$

Due to space constraints, we have only specified strategies to prove the theoretical aspects of the KIA model. Please contact us at 41872@student.hhs.se or 41885@student.hhs.se for utility values for other strategies.

B Appendix 2: Algorithms

B.1 Python Code: KIA

```
#-----------------IMORTING_PAKAGES----------------------------------------------
import networkx as nx
import random as rm
import matplotlib.pyplot as plt
import numpy as np
import math
np.set_printoptions(suppress=True)
import matplotlib.pyplot as plt
from sympy import Eq, var, solve
from collections import Counter
import os
import contextlib

#-----------------PARAMETERS----------------------------------------------------

#A = Adjacency matrix (Graph)

#alpha = Learning parameter (players' memory)

#iterations = number of iterations

#Delta = Parameter for knowledge transmissibility

#d = Parameter for knowledge complexity

#DATAGAP = gap in terms of iterations between a data point and another (when making
      graphs)

#replays = Number of times the KIA game is repeated

#-------THE_KIA_FUNCTION--------------------------------------------------------

def KIA(A, iterations, d, Delta, alpha, DATAGAP):

#--------SETUP------------------------------------------------------------------

    #number of players
    n = len(A)

    #list of players
    pl = 0
    Players = []
    while pl < n:
        Players.append(pl)
        pl = pl + 1

    #variables needed during the iteration process
    tot = 1.
    InitialBelief = 2.
    for x in range(0, n):
        globals()['CasePlayer'+str(x)] = -1
    EffectiveTransfer = np.array([[0.]*n]*n)
    PartialTransfer = np.array([[0.]*n]*n)
    PropagatedPlayers = []
    trust = 1

    #lists for plots
    PlotListTotalD = []
    TimeList = []
    for x in range(0, n):
        globals()['PlotList'+str(x)] = []
        globals()['PlotListU'+str(x)] = []

    #lists of edges
    Connections = []
    for x in range(0, n):
        for y in range(0, n):
            Connections.append(str(x)+" "+str(y))
    Connections = np.array(Connections)

    #strategy profile matrix
    StrategyProfile = np.array([[0.]*n]*n)


    #Variables needed in the analysis of results (counting the loops of players)
    LoopsChecker = np.array([[0.]*n]*n)
    BigLoopsChecker = np.array([[0.]*n]*n)
    LoopsChecker_vector = np.array([0.]*n)
    NONCONVERGINGChecker_vector = np.array([0.]*n)

    #Vector for k_i  (and a second one need to update the first one)
    KI = ([1./n]*n)
    KI_update = np.array([0.]*n)

    #Making Lists of out neighbours of every player and lists for their
        expectations
    for x in range(0, n):
        globals()['OutNeighbors_Player'+str(x)] = []
```

```
        for i in range(0, n):
            if A[x][i] == 1:
                (globals()['OutNeighbors_Player'+str(x)]).append(str(i))
        globals()['ExpectedReturns_Player'+str(x)] = [InitialBelief]*n
        for y in range(0, n):
            if str(y) not in (globals()['OutNeighbors_Player'+str(x)]):
                globals()['ExpectedReturns_Player'+str(x)][y] = 0

#--------ITERATIONS_BEGIN-------------------------------------------------------

    for i in range(1, iterations):

#--------FROM_EXPERIENCE_TO_STRATEGIES------------------------------------------

        StrategyProfile = np.array([[0.]*n]*n)
        for x in range(0, n):
            Possibilities = []
            MAX = max(globals()['ExpectedReturns_Player'+str(x)])
            for y in range(0, n):
                if MAX == (globals()['ExpectedReturns_Player'+str(x)])[y]:
                    Possibilities.append(y)
            Play = int(rm.choice(Possibilities))
            StrategyProfile.itemset((x, Play), 1.)

#--------PARTIAL_TRANSFER_OF_KNOWLEDGE_(STEP_1)---------------------------------
# (PHI function in formal model)

        PartialTransfer = np.array([[0.]*n]*n)
        EffectiveTransfer = np.array([[0.]*n]*n)
        PartialTransfer = PartialTransfer + StrategyProfile

#--------EFFECTIVE_TRANSFER_OF_KNOWLEDGE_(STEP_2)-------------------------------

        for x in range(0, n):
            Propagation = 0
            PropagatedPlayers = []
            PropagatedPlayers.append(x)
            globals()['CasePlayer'+str(Propagation)] = np.random.choice((Players),p
                =(PartialTransfer[x]))
            while Propagation < n:
                if PartialTransfer[x][x] == 1:
                    Propagation = n

                else:
                    if globals()['CasePlayer'+str(Propagation)] in
                        PropagatedPlayers:
                        Propagation = n

                    else:
                        if Propagation == n:
                            print("PROPAGATION ERROR")
                        else:
                            EffectiveTransfer.itemset((x, globals()['CasePlayer'+
                                str(Propagation)]), (Delta**Propagation))
                            PropagatedPlayers.append(globals()['CasePlayer'+str(
                                Propagation)])
                            Propagation = Propagation + 1
                            globals()['CasePlayer'+str(Propagation)] = np.random.
                                choice((Players),p=(PartialTransfer[globals()['
                                CasePlayer'+str(Propagation-1)]]))

            EffectiveTransfer.itemset((x, x), 1)

#--------KI_AND_UTILITY_MECHANISM-----------------------------------------------

        Utility_update = np.array([0.]*n)
        SUM = 0
        for x in range(0, n):
            for y in range(0, n):
                Utility_update[y] = Utility_update[y] + (KI[x] * EffectiveTransfer[
                    x][y])
                SUM = SUM + (KI[x] * EffectiveTransfer[x][y])

        KI_update = np.array([((1-d)/n)]*n)
        for x in range(0, n):
            KI_update[x]=KI_update[x]+((d*(Utility_update[x])/SUM))

        for x in range(0, n):
            Utility_update[x]= Utility_update[x] - KI[x]
            KI[x]=KI_update[x]
            Utility_update[x]= Utility_update[x] + KI[x]*kkk

#--------SETTING_EXPERIENCE-----------------------------------------------------

        for x in range(0, n):
            PlayerXPassedTo = np.random.choice((Players),p=(PartialTransfer[x]))
            (globals()['ExpectedReturns_Player'+str(x)])[PlayerXPassedTo] = round
                (((alpha*(Utility_update[x])) + (((globals()['
                ExpectedReturns_Player'+str(x)])[PlayerXPassedTo])*(1-alpha))),
                10)

        if i % int((iterations)/20) == 0:
            trust = trust + 1
            for x in range(0, n):
                for y in range(0, n):
                    if (globals()['ExpectedReturns_Player'+str(x)])[y] != 0:
```

```
                    (globals()['ExpectedReturns_Player'+str(x)])[y] = (globals
                        ()['ExpectedReturns_Player'+str(x)])[y] + (
                        InitialBelief/(2**trust))

#--------EXTRA_(PLOTS_AND_COUNTERS)-------------------------------------------

        tot = (sum(Utility_update))
        Onepercent = iterations//10
        if i % Onepercent == 0:
            print(str(i//Onepercent*10)+"%")

        if i % DATAGAP == 0:
            TimeList.append(i)
            for x in range(0, n):
                (globals()['PlotList'+str(x)]).append(KI[x])
                (globals()['PlotListU'+str(x)]).append(Utility_update[x])
            PlotListTotalD.append(tot)

#-----------RESULTS--------------------------------------------------------------

    for x in range(0, n):
        print("Player "+str(x))
        print('ExpectedReturns_Player')
        print(globals()['ExpectedReturns_Player'+str(x)])
        print('OutNeighbors_Player')
        print((globals()['OutNeighbors_Player'+str(x)]))
        for y in range(0, n):
            StrategyProfile.itemset((x, y), round((StrategyProfile[x])[y], 4))
    print('KI')
    print(KI)
    print(sum(KI))
    print('StrategyProfile')
    print(StrategyProfile)

    global equilibrium

    equilibrium = round(tot, 2)

    print(A)

    if DATAGAP < iterations:

        for x in range(0, n):
            plt.plot(TimeList, (globals()['PlotList'+str(x)]))
        plt.show()

        for x in range(0, n):
            plt.plot(TimeList, (globals()['PlotListU'+str(x)]))
        plt.show()

        plt.plot(TimeList, PlotListTotalD)
        plt.show()

        Connections = np.array([0]*n)
        for x in range(0, n):
            Connections.itemset((x), (len(globals()['OutNeighbors_Player'+str(x)]))
                )
        plt.scatter(Connections, KI)
        plt.show()


#-----------COUNTING_LOOPS-------------------------------------------------------

    NONCONVERGING = []
    LOOPS = []
    NONCONVERGING_Players = 0
    for x in range(1, n+1):
        #global (globals()['N_'+str(x)+'_Player_Loops'])
        globals()['N_'+str(x)+'_Player_Loops'] = 0



    for x in range(0, n):
        for y in range(0, n):
            NONCONVERGINGChecker_vector[x] = NONCONVERGINGChecker_vector[x] +
                StrategyProfile[y][x]
            BigLoopsChecker.itemset((x,y), StrategyProfile[x][y])
        if NONCONVERGINGChecker_vector[x] == 0:
            NONCONVERGING_Players = NONCONVERGING_Players + 1
            NONCONVERGING.append(x)
            for y in range(0, n):
                BigLoopsChecker.itemset((x,y), 0)

    for x in range(0, n):
        for y in range(x, n):
            if NONCONVERGINGChecker_vector[y] != 0:
                LoopsChecker.itemset((x,y), StrategyProfile[x][y] + StrategyProfile
                    [y][x])
            else:
                BigLoopsChecker.itemset((x,y), 0)

    for x in range(0, n):
        if LoopsChecker[x][x] == 2:
            LoopsChecker.itemset((x,x), 0)
            BigLoopsChecker.itemset((x,x), 0)
            LOOPS.append(str(x))
```

```
        for y in range(x+1, n):
            if LoopsChecker[x][y] == 2:
                LOOPS.append(str(x)+" "+str(y))
                LoopsChecker.itemset((x,y), 0)
                BigLoopsChecker.itemset((x,y), 0)
                BigLoopsChecker.itemset((y,x), 0)
        LoopsChecker_vector.itemset(x, sum(LoopsChecker[x]))

    L = 0


    LOOP = []
    SAME_LOOP = False
    while L < n:
        if sum(BigLoopsChecker[L]) == 0:
            if SAME_LOOP == False:
                L = L + 1
            else:
                if L in LOOP:
                    while len(LOOP) > 0:
                        if L == LOOP[0]:
                            L = 0
                            LOOPS.append(' '.join(str(e) for e in LOOP))
                            LOOP = []
                            SAME_LOOP = False
                        else:
                            NONCONVERGING.append(LOOP[0])
                            del LOOP[0]
                            NONCONVERGING_Players = NONCONVERGING_Players + 1
                else:
                    for x in range(0, len(LOOP)):
                        NONCONVERGING_Players = NONCONVERGING_Players + 1
                        NONCONVERGING.append(LOOP[x])
                    LOOP = []
                    SAME_LOOP = False
                    L = 0


        elif sum(BigLoopsChecker[L]) == 1:
            if SAME_LOOP == False:
                LOOP = [L]
                L = np.random.choice((Players),p=(BigLoopsChecker[L]))
                BigLoopsChecker.itemset((int(LOOP[-1]), L), 0)
                SAME_LOOP = True
            else:
                LOOP.append(L)
                L = np.random.choice((Players),p=(BigLoopsChecker[L]))
                BigLoopsChecker.itemset((int(LOOP[-1]), L), 0)
        else:
            print("LOOPS ERROR")

    print("NON-CONVERGING: " + str(NONCONVERGING))
    print("LOOPS: " + str(LOOPS))
    for x in range(0, len(LOOPS)):
        Dimension = (LOOPS[x]).count(" ") + 1
        globals()['N_'+str(Dimension)+'_Player_Loops'] = globals()['N_'+str(
            Dimension)+'_Player_Loops'] + 1


    print('NON-CONVERGING Players: '+ str(NONCONVERGING_Players))
    TotalPlayersinLOOPS = 0
    for x in range(1, n+1):
        print(str(x)+' Player Loops: '+ str(globals()['N_'+str(x)+'_Player_Loops'])
            )
        TotalPlayersinLOOPS = TotalPlayersinLOOPS + ( x * int(globals()['N_'+str(x)
            +'_Player_Loops']))

    print("TotalPlayersinLOOPS: " + str(TotalPlayersinLOOPS))

#-----------THE_END-------------------------------------------------------------
```

B.2 Python Code: TESTER

```
#-----------------IMORTING_PAKAGES----------------------------------------------

import networkx as nx
import random as rm
import matplotlib.pyplot as plt
import numpy as np
import math
np.set_printoptions(suppress=True)
import matplotlib.pyplot as plt
from sympy import Eq, var, solve
from collections import Counter
import os
import contextlib

#-----------------PARAMETERS----------------------------------------------------

#A = Adjacency matrix (Graph)

#alpha = Learning parameter (players' memory)

#iterations = number of iterations

#Delta = Parameter for knowledge transmissibility

#d = Parameter for knowledge complexity

#DATAGAP = gap in terms of iterations between a data point and another (when making
      graphs)

#replays = Number of times the KIA game is repeated


#-------THE_TESTER_FUNCTION-----------------------------------------------------

# This function replays the KIA function a predetermined amount of times

def Tester(A, iterations, d, Delta, alpha, DATAGAP, replays):

#--------SETUP------------------------------------------------------------------

# Setting basic necessary variables
    global results
    results = []
    r = 0

# Number of players
    n0 = len(A)

# Lists for plots
    PlotLOOP = []

# Variables needed to count loops
    PLAYERSINLOOP = 0
    for x in range(1, n0+1):
        globals()['TN_'+str(x)+'_Player_Loops'] = 0

#--------GAME_REPETITIONS-------------------------------------------------------

    while r < replays:

        #the function is run in a special environment in order to
        #avoid printing results at every repetition of the game
        with open(os.devnull, "w") as f, contextlib.redirect_stdout(f):
            KIA(A, iterations, d, Delta, alpha, DATAGAP)

        #results are saved
        results.append(equilibrium)

        #The number of the game is printed in order to keep track
        #of the progresses of the function
        print('GAME: '+str(r+1))

        #Store the count of players in loops
        for x in range(1, n0+1):
            for y in range(0, (globals()['N_'+str(x)+'_Player_Loops'])):
                PlotLOOP.append(x)

        #New game!
        r = r+1

#-----------PRINTING_RESULTS----------------------------------------------------

    plt.hist(results)
    plt.show()
    plt.hist(PlotLOOP)
    plt.show()
    print()
    for x in range(1, max(PlotLOOP)+1):
        print(str(x) + "-player loop: " + str(PlotLOOP.count(x)))
        PLAYERSINLOOP = PLAYERSINLOOP + (PlotLOOP.count(x)*x)
```

```
    print("PLAYERS IN LOOP: "+ str(PLAYERSINLOOP))
    print("NON CONVERGING PLAYERS: "+ str((replays * n0) - PLAYERSINLOOP))

#-----------THE_END-------------------------------------------------------------
```