UNIVERSITÀ DEGLI STUDI DI MILANO

DIPARTIMENTO DI FILOSOFIA
"PIERO MARTINETTI"

Corso di Laurea Magistrale in Scienze Filosofiche

A LOGIC FOR PARANOID AGENTS

Tesi di laurea

Presentata da:
Lorenzo Prandi

Relatore:
Prof.
Giuseppe Primiero

Anno Accademico 2018-2019

Ringrazio il Prof. Giuseppe Primiero per tutto il tempo che mi ha dedicato nella stesura di questa tesi, senza il suo aiuto questo lavoro non avrebbe potuto prendere forma. Ringrazio anche la mia famiglia per il sostegno incondizionato. Un ringraziamento speciale va alla mia fidanzata Claudia per essermi sempre stata vicino. Infine ringrazio gli amici della Tana Club.

Abstract

In this work we formulate a logic that mimics the behavior of conspiracy theorists, then we implement it in a multi-agent simulation. The logic $\texttt{(un)SecureND}^{\texttt{sim*}}$ is equipped with a proof-theory and relational semantics in which negative trust relations are defined formalising the attitude of paranoid agents. The experimental analysis aims at analysing consensus reaching transmissions in networks of agents with the presence of conspiracy theorists, and the ability of the latter to induce the spread of potentially false information. We consider in particular scale-free networks, in order to model real case scenarios, such as social networks.

Contents

Chapter 1

Introduction

Conspiracy theories are a worldwide phenomenon. A study showed that 1/3 of Americans believed that the U.S government either assisted or tacitly allowed the September 11th attacks (Oliver & Wood, 2014). Soon after the bombings in Bali, a survey by `indonesiadetik.com` showed that many believed that the CIA played an active role in the terrorist attack (Mashuri & Zaduqisti, 2014). Before the referendum for the withdrawal of the United Kingdom from the European Union, 46% of UK voters supporters of Brexit believed that the vote would be rigged (Douglas et al., 2019). Not only they are spread all over the word, but they can be easily found everywhere; it is hard to keep track of how many websites promote conspiracy theories and they are the topic of movies and songs.

Some of them are peaceful and extravagant, others are harmful, linked with racism and prejudices (Douglas et al., 2019). In the most extreme cases they are correlated with terrorism. A tragic example is the Christchurch mosque shootings where the thirty-year-old Brenton Tarrant killed 51 people. His actions were boosted by a "neonazi" propaganda in which fake news and conspiracy theories, such as the Kalergi Plan, were the main points (Macklin, 2019).

The literature on the argument is very broad and many disciplines have analyzed the phenomenon. However, not many contributions are from the domain of formal logic: the majority of them analyze the fallacies and errors in the arguments of the most famous conspiracy theories (Martínez et al., 1999).

For these reasons, the goal of this dissertation is to provide a formal analysis of an epistemic logic which mimics the behavior of conspiracy theorists, by providing formal rules to reason and predict their behavior. To account formally for this epistemic attitude, we characterize conspiracy theorists as agents who consider authorities unreliable sources of information, thereby rejecting all their contents, including previously held beliefs compatible with those. By following these considerations, we identify trust, and in particular distrust, as the proper relation to analyze the behaviour of a conspiracy theorist.

In chapter two it is presented the logic(s) from which the formal rules for the conspiracy theorist's behaviour are built on. Hence, it is introduced $\texttt{SecureND}^{\texttt{sim}}$ (Primiero et al., 2017), which is modelled specifically for simulation purposes, and its negation complete fragment (Primiero, 2019). Indeed, `(un)SecureND` combines access control operations of `read` and `write`, with a `trust` function for bridging

them through a consistency check operation on propositional contents; `mistrust` allows acceptance of contradictory incoming information through removal of previously held data; `distrust` preserves agent's local consistency through rejection of incoming inconsistent data. In chapter three, we modify this (family of) logic(s) to accommodate formal rules for a *paranoid agent*, mimicking the behaviour of a conspiracy theorist. The new rules of the logic $\texttt{(un)SecureND}^{\texttt{sim*}}$ are justified by showing their consistency with the current literature on conspiracy theories, regarding how they spread and how it is possible that a rational agent, exemplified in the model by an agent that is not "paranoid", could accept a conspiracy theory. The gist of the conspiracy theorist behaviour is condensed in the attitude which rejects any information sent from agents higher in the order relation, intuitively modelling the authority. The paranoid agent will not just distrust any such information, but also mistrust any previously held belief consistent with information incoming from an authority. This induces a continuous process of revision of the agent's own belief base, followed by an active process of information transmission. We model it both as a natural deduction system and a sound relational semantics.

In chapter four, to further extend our analysis, we present the design and the implementation of a NetLogo simulation of a network of agents, extending the experimental study offered in (Primiero et al., 2017) for sceptic and lazy agents. We consider total and small-world networks, and analyse a simplified scenario with only atomic information, where mistrust reduces to distrust. We present the results of the experimental analysis in chapter five, the aim is to study consensus reaching scenarios in the presence of conspiracy theorists, and the ability of such agents to induce the spread of potentially false information.

Prior to this, we expose hereafter some general remarks on conspiracy theories.

1.1 What is a conspiracy theory?

Determining whether an explanation of an event should be classified as a conspiracy theory is in most cases easy. Let us examine the many conspiracy theory about the death of the princess Diana, the ones concerning the assassination of the president John Fitzgerald Kennedy or the ones about the alleged fake moon landing; they are intuitively designated as conspiracy theories. However this is not always the case. For example the Watergate scandal could be designated intuitively as a conspiracy theory if we do not consider the proof of its truth; but if we do take it into account, it could also be designated simply as a conspiracy. Again, what about the conspiracy of all parents that make their children think that is Santa Claus who brings Christmas presents to them? Should it be considered a conspiracy theory? In order to disambiguate these cases and to understand more precisely the subject of this work, we present some definitions provided for the expression "conspiracy theory" in (Douglas et al., 2019), (Cohnitz, 2017) and (Sunstein & Vermeule, 2009). We do not fully commit to any of them, we just want to highlight the main factors that play a role in defining what a conspiracy theory is.

In (Douglas et al., 2019) conspiracy theories are defined by decomposing the term in its two components. In the first place, the term "conspiracy" is defined as a "secret plot by two or more powerful actors" (Douglas et al. (2019), p.3), in which the reference to powerful actors is added to exclude more ordinary crime. Finally,

"Conspiracy theories" are defined as

> "attempts to explain the ultimate causes of significant social and political events and circumstances with claims of secret plots by two or more powerful actors. [...] While a conspiracy refers to a true causal chain of events, a conspiracy theory refers to an allegation of conspiracy that may or may not be true." (Douglas et al. (2019), p.4)

Therefore, by embracing this definition the Watergate scandal should not be classified as a conspiracy theory, since its truth has been proven.

What about Santa Claus? In (Sunstein & Vermeule, 2009) conspiracy theory are defined as

> "an effort to explain some event or practice by reference to the machinations of powerful people, who attempt to conceal their role (at least until their aims are accomplished)." (Sunstein & Vermeule (2009), p.205)

This definition does not say anything about the truth or falsity of a conspiracy theory, so the Watergate scandal should be considered a conspiracy theory. However, in (Sunstein & Vermeule, 2009) it is clarified that true conspiracies, together with the ones that are not harmful, are not considered during its analysis of the phenomenon. Hence, the Santa Claus "conspiracy" is a conspiracy theory, but it is something we should not be worried about, since it is not harmful.

A more broad definition of conspiracy theories is given in (Cohnitz, 2017)

> "A conspiracy theory is an explanation that cites agents acting together in secrecy as a salient cause." (Cohnitz (2017), p. 16)

According to it, the crucial feature of conspiracy theories is the reference to a secret plot. Other factors such as the presence of a malicious plan or the involvement of powerful people are not necessary to designate an explanation as a conspiracy theory. To show the validity of this definition, Cohnitz (2017) presents the example of the Paul McCartney's conspiracy theory, which does not include any malicious plan. Indeed, according to it, the Beatles's bassist was substituted with a doppelganger after his alleged death.

1.2 Conspiracy Theories and Untrustworthiness

The rules presented in chapter 3 try to formalize the different possible trust relations in the context of information transmission between an agent defined as standard, more prone to trust what it is told to her, and an agent defined as paranoid, who should mimic the behaviour of a hardcore conspiracy theorist. Why trust should be the right relation to model the behaviour of a conspiracy theorists? Low trust levels towards institution and powerful people, or even low interpersonal trust, is correlated by many studies with the endorsement of conspiracy theories, for example see Douglas et al. (2019); Sunstein & Vermeule (2009); Abalakina-Paap et al. (1999); Bruder et al. (2013); Parsons et al. (1999); Mashuri & Zaduqisti (2014); Miller et al. (2016); Goertzel (1994).

Douglas et al. (2019) explain how political scandals diminish trust of people toward the government and how this eventually translates in the endorsement of conspiracy theories. Similarly, Sunstein & Vermeule (2009) describe a process playing between the endorsement of conspiracy theories and distrust: accepting conspiracy theories means also to accept an ubiquitous distrust in all the knowledge-producing institutions. In (Goertzel, 1994), a survey of 348 residents of southwestern New Jersey is analysed to show that the Belief in Conspiracies, mostly for people belonging to a minority, was correlated with a three-item scale of trust based on whether the people surveyed trusted the police, their neighbors or their relatives. Similar conclusions are sustained in (Mashuri & Zaduqisti, 2014), a statistical study among Indonesian Muslims on the widespread conspiracy theory about the involvement of Westerns people behind terrorism in Indonesia. They found that the belief in this conspiracy was predicted by distrust toward western people, group incompatibility and Islamic identification. (Miller et al., 2016), a study on ideological involvement of conspiracy theory between conservatives and liberals supporters in the US, found that for the conservatives high trust levels, as opposed to political knowledge, was a mitigating factor to the endorsement of conspiracy theories. Indeed, seeing the world as a trustworthy place makes it difficult to believe that political rivals are plotting against you.

Along these lines, the idea in the present work is to introduce an agent who does not accept information when it comes from agents above her in a hierarchy, as a consequence of the low trust levels that a conspiracy theorist hold towards powerful agents suspected of being malicious and evil. By formalizing the conspiracy theorist attitude with a logic to reasons about trust, we are trying to mimic this central factor that is in play when a conspiracy theory is accepted or even formulated.

1.3 Endorsement of Conspiracy Theories

In the present work, we account for two different views, often characterized as mutually exclusive, on the endorsement of conspiracy theories. According to the first view the endorsement of conspiracy theories depends on some special personal traits (see Mashuri & Zaduqisti (2014); Goertzel (1994); Oliver & Wood (2014); Bruder et al. (2013); Miller et al. (2016)).

Bruder et al. (2013) proposes the idea that the endorsement of conspiracy theories depends on personal traits, based on the discovery of people's tendency to believe in more conspiracy theories when the first one is endorsed. The different personal traits that predict the acceptance of conspiracy theories define a *conspiracy mentality*. In particular, some personal traits, proven to be related with the conspiracy mentality in the article, are anomia, paranoia and schizotypy, paranormal belief and openness to experience. Similarly, in (Miller et al., 2016) a particular profile, both knowledgeable about politics and lacking in trust, is identified as the most prone to ideologically motivated conspiracy endorsement. It is also showed that the endorsement of conspiracy theories are strongest in people with an ideological worldview compatible with a particular conspiracy theory, with the motivation to protect that worldview also by accepting a conspiracy theory and with low trust level.

These are some of the personal traits proposed. We do not want to take part in

the discussion on which are the right ones, but just point out that the paranoid agent (see chapter 3 is compatible with the view that relates the endorsing of conspiracy theories with specific characteristics. In our case, it is just considered the low trust level of the paranoid agent towards the authorities perceived as unreliable sources of information.

Instead, according to the second view, the endorsement of conspiracy theories does not require particular predispositions but it is due to the lack of appropriate information (see Sunstein & Vermeule (2009); Madsen et al. (2017)). For example, in (Sunstein & Vermeule, 2009) the endorsement of a conspiracy theory is due to a *crippled epistemology* consisting of limited informational sources. Indeed, accepting a conspiracy theory is the effect of belonging to an isolated group or networks only exposed to skewed information. In these circumstances, adhering to a conspiracy theory is not irrational from the standpoint of the isolated group; it appears unjustified to people that have more information available. It is maintained that this condition is related to how beliefs are acquired: most of times we have to rely on others, since acquiring direct information is not always possible. A similar position is sustained in (Madsen et al., 2017), where it is showed the possibility of growing a Bayesian conspiracy theorist through an Agent-Based Model with constraints on access to the total information. Therefore, according to Madsen et al. (2017), conspiratorial thinking does not require particular predisposition but can arise from a partial informational state.

This second view on the endorsement of conspiracy theories is compatible as well with the present work. In chapter 5 it is described the situation of information transmission where a non-paranoid agent, defined as standard (see chapter 3), could accept a conspiracy theory if it does not possess information that contradicts the transmitted biased information. Therefore, also in the present work, endorsing a conspiracy theory could be an effect of a lacking informational state.

Chapter 2

(un)SecureND

In the present chapter, it is introduced `(un)SecureND` and `(un)SecureND`$^{\texttt{sim}}$ on which the formal rules for the conspiracy theorist's behavior are built on. `SecureND` is a logic that models the relation of trust, an essential propriety in computational contexts where agents need to rely on external sources to execute decisions (Primiero et al., 2017). The logic resolves the problems of transitive trust and unintended trust multiplication. Normally trust is formalized as a first order relation, thus transitivity holds:

Example 1 (Trust Transitivity, (Primiero, 2019))**.** *If Alice trusts Bob and Bob trusts Carol, then Alice trusts Carol.*

However, this propriety is not always desirable, such as in security context where authorization should not be a consequence of transitivity.

Trust transitivity along with the question on how to define negative trust generate the problem of negative trust multiplication:

Example 2 (Negative Trust Multiplication, (Primiero, 2019))**.** *If Alice does not trust Bob and Bob does not trust Carol, then Alice trusts Carol.*

To avoid trust transitivity and negative trust multiplication, trust is not formalized as a first order relation but on a second-order propriety defined as follows:

> *(the profile of A) trusts (message from B)* (Primiero, 2019).

Hence, trust is intended as a bridging function between the content that can be read by an agent and the content that is allowed to write. In the end, trusting a message means to check its consistency with respect to the current profile. If the message is not consistent, the agent can either distrust, i.e reject the message, or mistrust, i.e reject a previously held data, depending on many factors, such as the behaviour of the agent or its ranking. The logic is provided with a proof system, an analyse of its structural properties, a relational semantics based on Kripke models and soundness and completeness results, see (Primiero, 2019).

The formal rules for the paranoid agent will be an extension of the system `(un)SecureND`$^{\texttt{sim}}$ presented in (Primiero et al., 2017). The research is about the role of trust in context where a hierarchical structure is in place, such as access control model. Trust is defined as either being a propriety of top-down communication,

where there is no need to verify information, and bottom-up transmission, where it is essential to validate information. The article offers a model of trust propagation in networks of ranked agents characterized by two different epistemic attitudes:

- *sceptic agents*: they pay an epistemic cost by performing a checking operation before trusting received information;
- *lazy agent*: they distrust the information when this is not consistent with their current knowledge." (Primiero et al. (2017), p.2)

In particular, `(un)SecureND`$^{\texttt{sim}}$ resolves the problem of contradictory information transmission in networks of agents with positive and negative trust. Positive trust is defined as a property of the communication between agents taking place when a message is transmitted bottom-up in the hierarchy, or as a result of a sceptic agent checking information. Instead, Negative trust is the result of rejecting received contradictory information. These two possibilities are associated with epistemic cost, and one of the goals of the article is to determine if it is more or less costly for a network resolving contradictory transmissions by rejecting information or by straight acceptance. The problem of the transmission of contradictory information is analyzed using two different but correlated methods; the introduction of the logic `(un)SecureND`$^{\texttt{sim}}$ and its implementation in NetLogo.

2.1 The logic `(un)SecureND`$^{\texttt{sim}}$

In this section, we present more specifically `(un)SecureND`$^{\texttt{sim}}$, "a natural deduction calculus whose rules define how agents can execute access operations on (atomic) formulas and their negations" (Primiero et al., 2017). The hierarchy of the epistemically characterized agents is modeled in an order relation. The operations associated with the agents are reading and writing; the former is equivalent to message receiving and the latter to message passing. These operations are formally developed in the following procedures:

- *verification*: it is required either by a top-down reading operation, i.e., when message passing is executed from below in the hierarchy; or by a reading operation performed by a sceptic agent;
- *falsification*: it is formulated as closure of verification under negation and it follows from reading contents that are inconsistent with the current knowledge of the receiver; or from a reading operation performed by a lazy agent;
- *trust*: is a function that follows from verification, when the content passed is consistent with the knowledge of the receiver;
- *distrust*: it is formulated as closure of trust under negation and it follows from falsification. (Primiero et al. (2017), p.6)

The syntax is defined as follows:

Definition 1 (Primiero et al. (2017)). *The syntax of* (un)$\texttt{SecureND}^{sim}$ *is defined by the following alphabet:*

$$V^{<} := \{\texttt{lazy}(\texttt{v}_\texttt{i}), \texttt{sceptic}(\texttt{v}_\texttt{i})\}$$
$$\phi^V := p^{v_i} \mid \neg\phi^V \mid Read(\phi^V) \mid Verify(\phi^V) \mid Write(\phi^V) \mid Trust(\phi^V)$$
$$\Gamma^V := \{\phi_1^{v_i}, \ldots, \phi_n^{v_i}\};$$

As explained in (Primiero et al., 2017), $V^{<}$ is the set of the epistemic agent, the apex indicates the order relation between the agents defined on $V \times V$, where $v_i < v_j$ means v_i is higher in the dominance relation than v_j; ϕ^V is a meta-variable for Boolean atomic formulae closed under negation and the agent's operations; Γ^{v_i} expresses a set of formulae signed by an agent $v_i \in V$ in which a formula ϕ^{v_i} is derivable, i.e. Γ^{v_i} is the *context* in which ϕ^{v_i} is derived. The empty context is denoted by $\cdot \vdash$.

Definition 2 (Judgement, (Primiero et al., 2017)). *A judgement* $\Gamma^{v_i} \vdash \phi^{v_j}$ *states that a formula* ϕ *is valid for agent* v_j *in the context* Γ *of formulas (including operations) of agent* v_i*.*

Hence, *judgments* define an operation executed by the agent on the left hand side of the derivability sign, on the formula typed by the agent on the right hand side (Primiero et al., 2017).

The system $\texttt{(un)SecureND}^{\texttt{sim}}$ is introduced in figure 2.1. The rules *Atom*, $\neg -$ *Intro*, $\Gamma -$ *formation* and *premise* are for inductive construction of a context Γ^{v_i}. Contexts are called *user-profile* and they are demanded to be consistent. $\Gamma^{v_i}; \phi^{v_j}$ indicates that the extension of the profile v_i with a formula ϕ from agent v_j is consistent; profile extensions that do not preserve consistency are not allowed. The rule *Read_down* expresses that messages can always be read downward. *Read_elim* is the corresponding elimination rule; when a message is read and preserve consistency, it can be owned. *Verify_high* claims that verification is required for messages coming from an agent lower in the dominance relation. Verification is also implemented in *Verify_sceptic*; messages are always verified if a sceptic agent is on the receiving side. The rule *Trust* establishes that a message can be trusted if it is verified and preserves consistency. *Write_trust* is the consequential elimination rule, and it states that a trusted message can be written. Verification is not implemented in two cases; when a message received is not consistent with the current context (*Unverified_contra*), and when a lazy agent is on the receiving side (*Unverified_lazy*). If verification is missing, the message is distrusted (*Distrust*), and the opposite message is written (*Distrust_elim*). In summary, Sceptic agents always verify messages, but they distrust them when they are not consistent with their currently held knowledge. Conversely, lazy agents never verify the messages, and they distrust them when incoming information is not consistent with their context.

Standard logical notions are defined as follows:

Definition 3 (Satisfiability, (Primiero et al., 2017)). *An* (un)$\texttt{SecureND}^{sim}$ *judgement* $\Gamma^{v_i} \vdash \phi^{v_i}$ *is satisfied if there is a derivation* D *and a branch* $D' \subseteq D$ *with a final step terminating with such a judgement.*

Definition 4 (Validity, (Primiero et al., 2017)). *An* (un)$\texttt{SecureND}^{sim}$ *judgement* $\Gamma^V \vdash \phi^V$ *is valid if there is a derivation* D *and for all branches* $D' \subseteq D$ *and for all agents* $v_i \in V$*, there is a final step terminating with such a judgement.*

$$\frac{}{p^{v_i} \in \phi^V}\ \text{Atom} \qquad \frac{p^{v_i} \in \phi^V}{\neg p^{v_i} \in \phi^V}\ \neg\text{-Intro}$$

$$\frac{\cdot \vdash \phi^{v_i}}{\phi^{v_i} \in \Gamma^{v_i}}\ \Gamma\text{-formation} \qquad \frac{\phi^{v_i} \in \Gamma^{v_i}}{\Gamma^{v_i} \vdash \phi^{v_i}}\ \text{premise}$$

$$\frac{\Gamma^{v_i} \vdash \phi^{v_i}}{\Gamma^{v_i}; \Gamma^{v_j} \vdash Read(\phi^{v_i})}\ read\_down$$

$$\frac{\Gamma^{v_i}; \Gamma^{v_j} \vdash Read(\phi^{v_i}) \qquad \Gamma^{v_j} \vdash \phi^{v_i}}{\Gamma^{v_j} \vdash \phi^{v_j}}\ read\_elim$$

$$\frac{\Gamma^{v_j} \vdash \phi^{v_j} \qquad \Gamma^{v_i} \vdash Read(\phi^{v_j})}{\Gamma^{v_i} \vdash Verify(\phi^{v_j})}\ verify\_high$$

$$\frac{\Gamma^{v_j} \vdash Read(\phi^{v_i}) \qquad \texttt{sceptic}(\texttt{v}_\texttt{j}) \in V}{\Gamma^{v_j} \vdash Verify(\phi^{v_i})}\ verify\_sceptic$$

$$\frac{\Gamma^{v_i} \vdash Verify(\phi^{v_j}) \qquad \Gamma^{v_i} \vdash \phi^{v_j}}{\Gamma^{v_i} \vdash Trust(\phi^{v_j})}\ trust$$

$$\frac{\Gamma^{v_i} \vdash Read(\phi^{v_j}) \qquad \Gamma^{v_i} \vdash Trust(\phi^{v_j})}{\Gamma^{v_i} \vdash Write(\phi^{v_j})}\ write\_trust$$

$$\frac{\Gamma^{v_i} \vdash \phi^{v_i} \qquad \Gamma^{v_i} \vdash Read(\neg\phi^{j})}{\Gamma^{v_i} \vdash \neg Verify(\neg\phi^{v_j})}\ unverified\_contra$$

$$\frac{\Gamma^{v_i} \vdash Read(\phi^{v_j}) \qquad \texttt{lazy}(\texttt{v}_\texttt{i}) \in V}{\Gamma^{v_i} \vdash \neg Verify(\phi^{v_j})}\ unverified\_lazy$$

$$\frac{\Gamma^{v_i} \vdash \neg Verify(\phi^{v_j})}{\Gamma^{v_i} \vdash \neg Trust(\phi^{v_j})}\ distrust \qquad \frac{\Gamma^{v_i} \vdash \neg Trust(\phi^{v_j})}{\Gamma^{v_i} \vdash Write(\neg\phi^{v_j})}\ distrust\_elim$$

Figure 2.1: The system (un)$\texttt{SecureND}^{sim}$

2.2 Meta-Theory

Proposition 1 (Primiero et al. (2017))**.** *Any successful* (un)$\texttt{SecureND}^{sim}$ *message-passing operation is a derivation tree including a Write-Read-(Verify-Trust)-Write series of sequents.*

From proposition 1, it follows that verification and trust are optional steps in a derivation if the message is received by a lazy agent. This means that in each derivation it can be counted the number of times a trust rules as occurred in it, this

measure is denoted as $\left|Trust(\phi^V)\right|_D$ (Primiero et al., 2017).

Theorem 1 (Primiero et al. (2017)). *$|Trust(\phi^V)|_D = |Verify(\phi^V)|_D$, for all $v_i \in V$.*

This result offers a resolution strategy when we may have to decide between two contradictory formulas; we accept the more trusted formula in the derivation tree. This procedure is defined as *Conflict Resolution by Trust Majority*:

Definition 5 (Conflict Resolution by Trust Majority, (Primiero et al., 2017)). *Given a derivation D_1 terminating in $\Gamma^{v_i} \vdash Write(\phi^{v_i})$ and a derivation D_2 terminating in $\Gamma^{v_j} \vdash Write(\neg\phi^{v_j})$, a new step holds which takes as premises $\Gamma^k \vdash Read(\phi^{v_i})$ and $\Gamma^k \vdash Read(\neg\phi^{v_j})$ respectively, and concludes $\Gamma^{v_k} \vdash \phi^{v_k}$ if and only if $|Trust(\phi^V)|_{D_1} > |Trust(\neg\phi^V)|_{D_2}$.*

Another strategy is obtained the opposite way by counting how many times a message has been distrusted in a derivation D, this measure is denoted by $\left|Distrust(\phi^V)\right|_D$ (Primiero et al., 2017).

Theorem 2 (Primiero et al. (2017)). *$|DisTrust(\phi^V)|_D = |\neg Verify(\phi^V)|_D$, for all $v_i \in V$.*

In this case we choose the least distrusted one. This strategy is defined as Conflict Resolution by Distrust Majority (Primiero et al., 2017):

Definition 6 (Conflict Resolution by Distrust Majority Primiero et al. (2017)). *Given a derivation D_1 terminating in $\Gamma^{v_i} \vdash Write(\phi^{v_i})$ and a derivation D_2 terminating in $\Gamma^{v_j} \vdash Write(\neg\phi^{v_j})$, a new step holds which takes as premises $\Gamma^k \vdash Read(\phi^{v_i})$ and $\Gamma^k \vdash Read(\neg\phi^{v_j})$ respectively, and concludes $\Gamma^{v_k} \vdash \phi^{v_k}$ if and only if $|DisTrust(\phi^V)|_{D_1} < |DisTrust(\neg\phi^V)|_{D_2}$.*

The following lemma, derived from definition 4, is fundamental for the experimental purposes:

Lemma 1 (Primiero et al. (2017)). *For each (un)SecureND^{sim} derivation D with a valid formula $\Gamma^V \vdash \phi^V$, there is a graph G that is unanimously labelled by ϕ.(Primiero et al., 2017),*

In addition, the following structural proprieties on (un)SecureND$^{\mathtt{sim}}$ are proven (Primiero et al., 2017):

Lemma 2 (Primiero et al. (2017)). *For a derivation D of (un)SecureND^{sim}, the value of $|Trust(\phi^V)|_D$ is directly proportional to the number of $verify\_high$ rule applications and the number of distinct $\mathtt{sceptic}(\mathtt{v_i}) \in V$ occurring as labels in the premises of the derivation.*

Lemma 3. *For a derivation D of (un)SecureND^{sim}, the value of $|\neg Trust(\phi^V)|_D$ is directly proportional to the number of $unverified\_contra$ rule applications and the number of distinct $\mathtt{lazy}(\mathtt{v_i}) \in V$ occurring as labels in the premises of the derivation.*

Lemma 4. *Given a (un)SecureND^{sim} derivation D, the formula $\Gamma^{v_i} \vdash \phi^{v_i}$ converges to validity in D and to full labelling in the corresponding graph G as a direct function of:*

- *the number of instances of the verify_high rule applications.*
- *the number of instances of the verify_sceptic rule applications, for each* $v_i \in V$.

where ϕ^{v_i} *occurs in the conclusion, and as an inverse function of:*

- *the number of instances of the unverified_contra rule applications.*
- *the number of instances of the unverified_lazy rule applications, for each* $v_i \in V$.

where ϕ^{v_i} *occurs in the first premise.*

2.3 Simulation and Experimental Analysis

Experimental setting

Primiero et al. (2017) offers an agent based *NetLogo* simulation, which implements algorithms based on `(un)SecureND`$^{\texttt{sim}}$. The purpose is to test experimentally the proprieties of knowledge distribution, depending on the epistemic attitude of the seeding agents and on the network topology. Networks of agents are represented by undirected graphs; nodes represent the epistemic agents, while edges transmissions between them.

Definition 7 (Graph, (Primiero et al., 2017))**.** *A network is an undirected graph* $G = (V, E)$*, with a set* $V = \{v_i, \ldots, v_n\}$ *of vertices representing our agents and a set* $E = \{e_{(i,j)}, \ldots, e_{(n,m)}\}$ *of edges, representing transmissions among them.*

The labelling of the vertices is defined as follows:

Definition 8 (Labelling, (Primiero et al., 2017))**.** *Each vertex* $v_i \in V$ *can be labelled by formulas as follows:*

- $v_i(p)$ *denotes a vertex labelled by an atomic formulas and expresses an agent* i *knowing* p*;*
- $v_j(\neg p)$ *denotes a vertex labelled by the negation of an atomic formula and expresses agent* j *knowing* $\neg p$*;*
- $v_k()$ *denoted a vertex with no label and expresses an agent* k *who does not hold any knowledge yet.*

A transmission channel is formalized as an edge between two labelled nodes, denoted by $e(v_i(p), v_j())$; in this case i transmits p to j. The case of a contradictory transmission is admissible, i.e. $e(v_i(p), v_j(\neg p))$, though it requires a resolution procedure.

The experiments are conducted in four different network topologies: total, linear, random and free scale. The order relation over the agents is preserved, and it becomes total or partial depending on the topology. In a *total network* each vertex has an edge connected to any other, and equal ranking is assigned to all the agents,

```
PROCEDURE Transmission(G)

G := (V, E)
FOR e(v_i(φ), v_j()) ∈ G
      IF ((v_j() ∈ sceptic) AND (random−float 1 =< 0.95)) OR ranking(v_j()) < ranking(v_i(φ))
         THEN Verify(e(v_i(φ), v_j())) AND G := G ∪ (v_j(φ))
      ELSEIF ((v_j() ∈ lazy) AND (random−float 1 =< 0.80))
         THEN Distrust(e(v_i(φ), v_j())) AND G := G ∪ (v_j(¬φ))
      ENDIFELSE
  ENDFOR

  FOR e(v_i(φ), v_j(), v_k(¬φ)) ∈ G
      SolveConflict(e(v_i(φ), v_j(), v_k(¬φ)))

  RETURN Trusted(G)
  ENDPROCEDURE
```

Figure 2.2: Algorithm for Simple Information Transmission

hence the order is total. In a *linear network* each vertex as an edge connected to the next vertex higher in the ranking, therefore the order relation is total. A *random network* is created making sure that each vertex is connected at least with another, though the order relation over the agents is partial. In a *scale-free network* the creation of edges follows a power-law degree distribution; newly added nodes tend to prefer vertices with a high number of links. The ranking is determined by a simple function $\frac{1}{|edges|}$. Scale-free world should depict real social networks cases (Primiero et al., 2017).

The randomly seeded contradictory information p and $\neg p$ spreads across the network. The diffusion of the message is guided by the algorithm *TRANSMISSION* (Primiero et al., 2017), which describes the relation between the agent who is sending the message and an agent receiving it.

Consider, for example, a sending agent being labeled by p and a receiver being not labeled yet. If the receiver is sceptic or if its ranking is lower than the one of the sender, it calls the *VERIFY* routine (Primiero et al., 2017) and it is labeled with p. From now on, the link between the two agents becomes of trust, and the number of trusted links is increased by one. If the receiver is lazy, it calls the *DISTRUST* routine (Primiero et al., 2017) and the new node is labeled by $\neg p$. In this case the subroutine *DISTRUST* increases by one the number of distrusted links.

```
PROCEDURE Verify(e(v_i(φ), v_j()))

set COSTTRUST+1
set TRUSTLINK e(v_i(φ), v_j(φ))
RETURN Trusted(G)
ENDPROCEDURE
```

Figure 2.3: Algorithm for Trust Costs Increase

```
PROCEDURE Distrust(e(v_i(φ), v_j()))

set COSTDISTRUST+1
set DISTRUSTLINK e(v_i(φ), v_j(¬φ))
RETURN Trusted(G)
ENDPROCEDURE
```

Figure 2.4: Algorithm for Distrust Costs Increase

The epistemic description of the agents is an approximation of the distinction between lazy and sceptic behaviour introduced in the logic. To offer a more realistic description of the agents, simulations are not only conducted on different networks topologies, but also on vary distributions of the epistemic attitudes with a semi-random implementation of the corresponding procedures:

1. **overly lazy network**: in this type of network, the proportion of sceptic nodes is set at 20%, with their confirmation rate at 5%, the latter expressing the proportion of such agents that will after all ask for verification;
2. **balanced network**: in this type of network, the proportion of sceptic nodes is set at 50% and their confirmation rate at 95%, to account for a 5% of random sceptic agents who decide not to ask for verification after all;
3. **overly sceptic network**: in this last type of network, the proportion of sceptic nodes is set at 80%, their confirmation rate at 100%, hence verification is always implemented. (Primiero et al., 2017)

When a node labelled by an atom p is linked to another node labelled by $\neg p$, the resolution routine *SOLVECONFLICT* (Primiero et al., 2017) is called. As for the logic, two different resolution procedures are offered. The first one counts the number of links with nodes labelled by p and the number of links with nodes labelled by $\neg p$. Then, it sums them considering their overall rankings, obtaining respectively the value of *ScoreP* and *Score*$\neg p$. If the former is greater than the latter, the new node is labelled by p or by $\neg p$ otherwise. The second one selects the least distrusted information, or it randomly chooses the label if the atoms are equally distrusted.

PROCEDURE `SolveConflict`$(e(v_i(\phi), v_j(), v_k(\neg\phi)))$

TotalP $= \#(V_i(\phi), V_j())$
Total$\neg$ P $= \#(V_k(\neg\phi), V_j())$
ScoreP $= 1/\text{ranking}(V_i(\phi)) + (\text{TotalP}/\#V)$
Score$\neg P = 1/\text{ranking}(V_i(\neg\phi)) + (\text{Total}\neg\ \text{P}/\#V)$

FOR $e(v_i(\phi), v_j(), v_k(\neg\phi)) \in$ Transmission(G)
  IF (ScoreP $>$ Score $\neg P$)
    THEN $G := G \cup v_k(\phi)$
  ELSEIF (ScoreP $=$ Score$\neg P$)
    THEN $G := G \cup v_k(\neg\phi)$
  IF (random$-$**float** $1 >= 0.5$)
    THEN $(v_k(\phi))$
    ELSE $(v_k(\neg\phi))$
  ENDIF
ENDFOR

RETURN Trusted(G)
ENDPROCEDURE

Figure 2.5: Algorithm for Conflict Resolution by Trust Majority

```
PROCEDURE SolveConflict2(e(v_i(φ), v_j(), v_k(¬φ)))

let d1 #DISTRUSTLINK e(v_{i,...n}(φ), v_j())
let d2 #DISTRUSTLINK e(v_{k,...m}(¬φ), v_j())

IF (length d1 > length d2)
   THEN G := G ∪ (v_j(¬φ)) AND Distrust(e(v_i(φ), v_j(¬φ)))
ENDIF

IF (length d1 < length d2)
    THEN G := G ∪ (v_j(φ)) AND Distrust(e(v_k(¬φ), v_j(φ)))
ENDIF

IF (length d1 = length d2)
   IF (random−float 1 =< 0.5)
   THEN G' := G ∪ (v_j(¬φ)) AND Distrust(e(v_i(φ), v_j(¬φ)))
   ELSE G' := G ∪ (v_j(φ)) AND Distrust(e(v_k(¬φ), v_j(φ)))
   ENDIFELSE
ENDIF
ENDPROCEDURE
```

Figure 2.6: Algorithm for Conflict Resolution by Distrust Majority

Experimental results

The procedure *Clearp* (Primiero et al., 2017) is necessary to obtain reliable experimental results; *Clearp* eliminates all the labels from the graph, but preserves rankings and trusted links for the next run of the simulation. In this sense, this procedure allows to run the experiments on a*memory-preserving* network. This is an essential feature since, as it has been showed in (Primiero et al., 2017), *memory-less* networks are not a reliable sources to run experiments.

By running multiple simulations on this setting, the article analyses consensus, costs and the ranking of the seeding nodes. *Consensus* is reached when a graph become unanimous. Primiero et al. (2017) showed that network topology affect directly consensus: total networks reach consensus more often, followed by scale-free, linear and random. As for the different configurations, an overly sceptic configuration performs better, while overly lazy configurations perform the worst. In fact, the clustering of lazy nodes is inversely proportional to the construction of trusted edges, and disadvantageous to consensus reaching transmission. Moreover, increasing the number of nodes while keeping constant the proportion between lazy and sceptic, increases the probability of clusters of lazy nodes, which reduces the number of trusted edges, causing the decrease of the number of runs in which consensus is reached.

Networks with trust and distrust present a different correlation between size and the number of times consensus is reached; small networks reach consensus more often. Furthermore, a distrust routine has a strong impact on the ability of the network to reach consensus in presence of contradictory information. The experimental analysis on consensus supports empirically the structural proprieties of $\texttt{(un)SecureND}^{\texttt{sim}}$. Total networks corresponds to derivation with the maximal number of branches; one for each pair of agent (v_i, v_j) appearing respectively in the premises and in the conclusions. Overly sceptic networks corresponds to derivations where more instances of the $verify_s ceptic$ rule are applied. Hence, as stated in lemma 4, the convergence of a formula to validity and the full labelling of the correspond-

ing graph resulting in consensus are maximal. Conversely, overly lazy networks corresponds to derivations where more agents implement the *unverified_lazy* rule.

Epistemic cost is the computational expense required to perform verification and distrust operations (Primiero et al., 2017), in the calculus it corresponds to the application of the rules *verify_high*, *verify_sceptic*, *unverified_contra* and *unverified_lazy*. As a consequence of the proprieties of the proof system, the epistemic cost for trust is higher than that of distrust, since verification requires more conditions. Random networks are the most epistemically expensive, while linear networks are slightly better than scale-free ones. An analysis on the different possible configurations of scale-free networks has showed that an overly sceptic network is the more expensive, and the difference in cost increases for larger overly lazy and overly sceptic networks. The gap is restricted between balanced and overly lazy networks. Therefore, if we want to choose between epistemic cost and consensus, large lazy network should be preferred. The average number of trusted and distrusted links grows in parallel, while the related cost decreases across the different topologies. However, the proportion is not linear: trust propagates at a much higher rate than distrust, which means that trust is a more relevant propriety in information transmission.

The analysis on epistemic cost is consistent with the theoretical proprieties of the calculus. Lemma 2 states that trust instances are proportional to the application of the *Verify_high* rule. In the simulations, this propriety is reflected in the observation that random network are more expensive than linear, because for the latter the high number of transitively valid transmission means fewer instances of the *verify_high* rule.

As for *Ranking*, see Primiero et al. (2017), there is no strict correlation between the ranking of the seeding node and consensus. In this respect, an overly sceptic scale-free network has the higher probability to reach unanimity when the seeding node is high in the ranking.

An analysis on scale-free networks (Primiero et al., 2017) showed that there is a strict relation between the proportion of lazy nodes and the distrust behaviour; more lazy nodes means a higher distrust value. As for a fully sceptic network, the value of distrust is related to the presence of contradictory information, hence distrust cost could be considered as a parameter of contradiction diffusion. In addition, there is a strict correlation between the final distribution of distrust values with the initial condition of the networks; we have a maximal decrease when the seeding nodes are sceptic as opposed to lazy, while the minimal values are stable. This means that the effects of distrust across the network is less influenced by the role of the agents transmitting information than the attitude of the agent receiving it.

The main results of the experiments can be summarized as follows (Primiero et al., 2017):

- The sceptic attitude is more likely to reach consensus, but it is also more expensive then the lazy one. Though, the former should be pursued if the maximization of consensus is the scope, and the lazy one if the minimization of costs is the goal.

- By comparing trust and distrust, it has emerged that the former is a better way to transmit information.

- Even in network with only sceptic agents, the presence of contradictory information generates distrust.

Chapter 3

A logic for Paranoid Agents

In this chapter, we introduce a logic for a *paranoid agent*. The purpose is to extend the current system with an additional epistemic attitude realized by a set of formal rules to mimic the behaviour of a conspiracy theorist. The interest in adding such rules in a logic that models the relation of trust is supported by findings in the literature that relate the endorsement of conspiracy theory with low trust level towards institutions and powerful people, allegedly being involved in nefarious plots (see chapter 1). Accordingly, the idea in the present work is to introduce an agent who does not accept consistent information when it comes from above in the hierarchy, as a consequence of the low trust levels that a conspiracy theorist holds towards powerful agents suspected of being malicious and evil.

3.1 Paranoid Agents and Standard Agents

The paranoid agent, similarly to a sceptic agent, will always need to trust messages; either the messages are coming from an agent lower in the dominance relation, or the information is transmitted from the top. The choice to base the paranoid behaviour on the sceptic one is not arbitrary. As it is argued in (Cohnitz, 2017, p.18):

> "The picture of the lazy, gullible, ignorant conspiracy theorists seems inappropriate for many cases. They are sceptics, they look at the information they receive via mainstream channels more critically than others."

Moreover, (Cohnitz, 2017) holds that conspiracy theorists don't lazily believe what they are told, but they do more: they look out for the extra information that is ignored by the mainstream, at the price of distrusting the relevant experts. Their vice lies in an overvaluation of their competence in judging over issues they don't possess the necessary preparation for, which leads them to claim bizarre false beliefs.

Accordingly to this view, the paranoid agent will implement an anomalous form of the sceptic behaviour: when it receives a consistent message coming from the top of the hierarchy, it will not trust it, instead it will trust the negation of a previously held formula, from which the new message can be derived. By following this procedure, the paranoid agent, besides not trusting a consistent message sent by an agent higher in the ranking, will also review its context in a way that will remove any information consistent with the information transmitted by the allegedly malicious

$$\frac{\Gamma^A \vdash Read(\psi_i^B) \to \bot \qquad \Delta^A; \psi_i^B : profile}{\Delta^A; \psi_i^B \vdash \neg Trust(\neg\phi_j^A)}\ MTrust\_I, \text{ With } \Delta^A \subseteq \Gamma^A \ni \phi_j^A \to \neg\psi_i^B$$

$$\frac{\Delta^A; \psi_i^B \vdash \neg Trust(\neg\phi_j^A) \qquad \Gamma^C; \psi_i^B : profile}{\Delta^A; \Gamma^C \vdash Trust(\psi_i^B)}\ MTrust\_E, \text{ for every } C \leq B$$

Figure 3.1: *Mtrust* introduction and elimination rules

authority. To obtain this result, the paranoid agent will implement an eccentric form of mistrust. In (Primiero, 2019), mistrust is formalized by the introduction and elimination rules presented in Figure 3.1

The introduction rule states that agent A, when it reads a formula ψ_i^B inconsistent with its current profile Γ^A from agent B, identifies a subset Δ^A of its context Γ^A by removing any contradictory formula ϕ_j^A, so that the profile is still valid when ψ_i^B is added. In some cases, in order to write ψ_i^B, more formulae are required to be removed through several iterations of the rule. The elimination rule establishes that Δ^A, before trusting the new message, should identify the profiles higher in the ranking consistent with the new formula.

The new rule to mimic the paranoid behaviour will be a variation of the *MTrust_I* rule. Indeed, the paranoid agent will not look at all its formulae that generate inconsistencies with the incoming formula, she will rather look at all its formulae from which the new one can be derived, and she will accept their negation. Considering that the rule for the paranoid agent will lead to trust information when is coming from both the top or the bottom of the hierarchy, as the *verify_sceptic* rule, the sceptic behaviour will not be implemented in the new proof system, as it becomes a specific form of the paranoid agent. Moreover, the exclusion of an epistemic attitude will keep the system simple, an important feature for the experimental purposes too.

To analyze the interaction of the paranoid agent with others, we will make explicit an epistemic behaviour already implicitly formalized in (Primiero et al., 2017) by the *verify_high* rule and the *unverified_contra* rule. Hence, we will introduce an agent who requires to trust messages coming from an agent lower in the hierarchy before owning them, and who distrusts contradictory information. We will refer to it as the standard agent. However, in the new system, its behaviour will be refined thanks to the the *Mtrust* rule introduced in (Primiero, 2019). Thus, the standard agent will mistrust messages that are not consistent with information provided by the authority. Since the standard agent is made explicit, and since it is a better candidate to analyze its interaction with the paranoid agent, the lazy attitude will not be implemented, meaning that the *unverified_lazy* rule will be neglected in the new system. Another difference from the previous setting is that connectives are introduced. They will be an adapted version of the introduction and elimination rules for connectives presented in (Primiero, 2019).

3.2 The Logic (un)SecureND^{sim*}

The new logic (un)SecureND^{sim*} preserves from the previous system presented in Chapter 2 the operations of reading, writing, trust and distrust for agent, while it differs for two different form of mistrust: one for the standard agent and one for the paranoid agent. Verification and falsification will not be implemented in the current setting, as the scope of the extended system is not to count the number of trusted formulae. However, since trust and verification, as distrust and falsification, were defined on a requirement of consistency, not implementing them will not cause major differences.

The syntax of the new logic is a modified version of the old one:

Definition 9. *The syntax of* (un)SecureND^{sim*} *is defined by the following alphabet:*

$$\mathcal{S}^{\leq} := \{\texttt{standard(I)}, \texttt{paranoid(I)}\}$$
$$\phi^{\mathcal{S}} := p^I \mid \neg\phi^I \mid \phi^I \rightarrow \psi^I \mid \phi^I \wedge \psi^I \mid \phi^I \vee \psi^I \mid \bot \mid Read(\phi^I) \mid$$
$$Write(\phi^I) \mid Trust(\phi^I)$$
$$\Gamma^I := \{\phi_1^I, \ldots, \phi_n^I\};$$

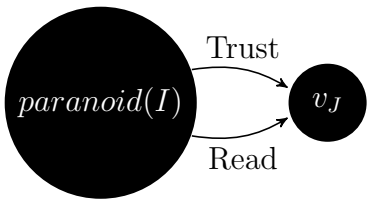
$\mathcal{S}^{\leq}$ is the set of epistemic agents, in the current setting $sceptic(v_i)$ is substituted with $paranoid(I)$, while $lazy(v_i)$ is replaced with $standard(I)$. The apex still indicates the order relation between the agents defined on $\mathcal{S}^{\leq} \times \mathcal{S}^{\leq}$, where $(I \leq J)$ means I is higher in the dominance relation than J, i.e. it occupies a relevance position in the group. $\phi^{\mathcal{S}}$ is a meta-variable for formulae, with their logical composition inductively defined by connectives under negation and the agent's access operations. The language includes $\bot$ to express conflicts by implication to contradiction. Γ^I expresses a set of formulae signed by an agent $I \in \mathcal{S}^{\leq}$ in which a formula ϕ^I is derivable.

The definition of judgment is still valid:

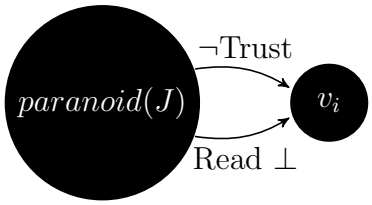
Definition 10 (Judgement, (Primiero et al., 2017))**.** *A judgement* $\Gamma^I \vdash \phi^J$ *states that a formula* ϕ *is valid for agent* J *in the context* Γ *of formulas (including operations) of agent* I*.*

Assuming $I \leq J$, valid transfers for the paranoid agent are summarised as follows by judgements of (un)SecureND^{sim*} :

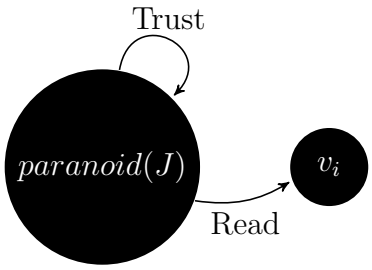
- If $paranoid(I)$, and $\Gamma^I \vdash Read(\phi^J)$, Then $\Gamma^I \vdash Trust(\phi^J)$: when a paranoid agent reads a formula coming from below, Trust is required. The same transfer is valid for $standard(I)$.



- Assuming $paranoid(J) \in \mathcal{S}^{\leq}$. If $\Gamma^J \vdash Read(\phi^I)$, and $\Gamma^J \vdash \neg\phi^J$, then $\Gamma^J \vdash \neg Trust(\phi^I)$: A paranoid agent does not trust inconsistent information. The same holds if the message is transmitted downward, in only this occurrence the transfer is valid also for $standard(I)$.



- If *paranoid*(J), and $\Gamma^J \vdash \mathit{Read}(\phi^I)$, Then $\Delta^J \vdash \mathit{Trust}(\neg\psi^J)$, for any ψ such that $\psi^J \vdash \phi^I$: when a paranoid agent reads a consistent formula coming from the top, it looks at all its formulae ψ^J from which ϕ^I is derivable, and it trusts its negation. Instead, if *standard*(J) is reading, the message is trusted.



The profile construction and operational rules for the new system are introduced in Figure 3.2. The rules *Atom*, $\Gamma -$ *formation* and *premise*, as in (Primiero et al., 2017), are for inductive construction of a context Γ^I. Contexts are still called *user-profile* and they are demanded to be consistent as well. $\Gamma^I; \phi^J$ indicates that the extension of the profile I with a formula ϕ from agent J is consistent; profile extensions that do not preserve consistency are not allowed.

The operational rules are an adapted version, for the current setting, of the rules that formulate closure under compositionality by logical connectives introduced in Primiero (2019). The meaning of the connectives are defined by a set of introduction rules and a set of elimination rules. As canonical in proof-theory, the former indicates a set of rules that establishes how a formula with connectives is obtained, the latter denotes a set of rules that express how the connectives are removed. $\perp$ is exploited to introduce $\neg$, as it expresses inconsistency of a profile, which induces admissibility of any formula (*ex falso sequitur quodlibet*). $\wedge$-I rule is the introduction rule for conjunction, it authorizes composition of formulas from distinct profiles; while the corresponding elimination rules, $\wedge - E_{1,2}$ rule, establish that each conjunct is derivable under the extended profile. The introduction rules for disjunction, $\vee - I\_1, 2$, claim that an extended profile can compose any valid formula from each profile; by the elimination rules,$\vee - E\_1, 2$, each formula derivable from each profile can also be derived from the extension of the profiles. $\rightarrow$-I is a form of *Deduction Theorem* and qualifies an inference from an extended profile as inference between formulae; instead, its elimination rule, $\rightarrow$-E, is a form of *Modus Ponens* which allows to recover such inference between formulae as profile extension.

The access rules are introduced in figure 3.3. These rules are strictly constrained by the behaviour of the agents, that causes in turn different inferences depending on the order of the agents.

As the system in (Primiero et al., 2017), *read* expresses that any agent is allowed to access any other agents message. Therefore, the rule is not constrained on the

$$\frac{}{p^I \in \phi^{\mathcal{S}}}\ \text{Atom} \qquad \frac{\Gamma^I \vdash \phi^I \to \bot}{\Gamma^I \vdash \neg\phi^I}\ \bot$$

$$\frac{\cdot \vdash \phi^I}{\phi^I \in \Gamma^I}\ \Gamma\text{-formation} \qquad \frac{\phi^I \in \Gamma^I}{\Gamma^I \vdash \phi^I}\ \text{premise}$$

$$\frac{\Gamma^I \vdash \phi^I \qquad \Gamma^J \vdash \psi^J}{\Gamma^I;\Gamma^J \vdash \phi^I \wedge \psi^J} \wedge -I$$

$$\frac{\Gamma^I;\Gamma^J \vdash \phi^I \wedge \psi^J}{\Gamma^I;\Gamma^J \vdash \phi^I} \wedge -E_1 \qquad \frac{\Gamma^I;\Gamma^J \vdash \phi^I \wedge \psi^J}{\Gamma^I;\Gamma^J \vdash \psi^J} \wedge -E_2$$

$$\frac{\Gamma^I;\Gamma^J \vdash \phi^I}{\Gamma^I;\Gamma^J \vdash \phi^I \vee \psi^J} \vee -I_1 \qquad \frac{\Gamma^I;\Gamma^J \vdash \psi^J}{\Gamma^I;\Gamma^J \vdash \phi^I \vee \psi^J} \vee -I_2$$

$$\frac{\Gamma^I;\Gamma^J \vdash \phi^I \vee \psi^J \qquad \phi^I \vdash \epsilon^K}{\Gamma^I;\Gamma^J \vdash \epsilon^K} \vee -E_1$$

$$\frac{\Gamma^I;\Gamma^J \vdash \phi^I \vee \psi^J \qquad \psi^J \vdash \epsilon^K}{\Gamma^I;\Gamma^J \vdash \epsilon^K} \vee -E_2$$

$$\frac{\Gamma^I;\psi^J \vdash \phi^K}{\Gamma^I \vdash \psi^J \to \phi^K} \to -I \qquad \frac{\Gamma^I \vdash \psi^J \to \phi^K \qquad \Gamma^I \vdash \psi^J}{\Gamma^I;\psi^J \vdash \phi^K} \to -E$$

Figure 3.2: The system (un)$\texttt{SecureND}^{sim}$: Profile construction and operational rules

order relation, as standard access control whereas $A \leq B$ means that agent A has access to the agent B content. In our setting $\leq$ represents an intuitive and informal concept of the power and the reputation of an agent.

Trust and *Dtrust* are the current version of the rule *verify_high* and *unverified_contra* presented in (Primiero et al., 2017). They concern messages transmitted upward, and they are valid both for the paranoid and the standard behavior. In particular, *Trust* expresses that messages coming from the bottom of the hierarchy are required to be trusted. Instead, *Dtrust* establishes that contradictory messages coming from below are distrusted.

Trust_low_std, *Mtrust_prd*, *Dtrust_prd* and *Mtrust_std* cover the instances of messages transmitted downward; in these occurrences the behaviour of the paranoid agent is nearly symmetrical to the standard one.

Trust_low_std and *Mtrust_prd* deal with the case where an agent lower in the hierarchy receives a consistent message. If the receiver is standard, it applies the *Trust_low_std*, an adjusted version of the *verify_sceptic* rule in (Primiero et al., 2017). This rule expresses that a standard agent trusts consistent messages coming from the top of the hierarchy.

Mtrust_prd establishes not only that a paranoid agent rejects a consistent message ϕ^I coming from an agent higher in the order relation, but also that she rejects its previously held formulae ψ^J, from which ϕ^I can be derived. In particular, the paranoid agent does so by identifying a subset Δ^J of its context Γ^J, which is still consistent with the removal of ψ^J and the addition of $\neg\psi^J$. The procedure should be iterated until the context has not any formula ψ^J such that $\psi^I \vdash \phi^I$. This is a modified version of the $MTrustI$ rule introduced in (Primiero, 2019).

Dtrust_prd and *Mtrust_std* cover the transmission of contradictory information coming from above. *Dtrust_prd* establishes that a paranoid agent distrust contradictory information coming from an agent higher in the hierarchy. However, if the contradictory information is received by a standard agent, *Mtrust_std* is applied. The rule states that agent J, when she reads a formula ψ^I inconsistent with its current profile Γ^J from agent I, identifies a subset Δ^J of its context Γ^I by removal of a formula ϕ^J, so that the profile is still valid when ψ^I is added. In some cases, in order to write ψ^I, more formulae are required to be removed through several iterations of the rule. This is the same rule for mistrust introduced in (Primiero, 2019).

Write_trust and *Distrust_elim* are the elimination rule respectively for the trust and distrust operations. *Write_trust* states that a trusted message can be written, whereas *Distrust_elim* claims that distrusting a message implies the contradictory. Finally *Derive* is the elimination rule for *Write_trust* and *Distrust_elim*, it claims that any written formula can be derived in a consistent profile.

The definition of Satisfiability (definition 3) and Validity (definition 4) are valid as well in the new system.

Definition 11 (Satisfiability, (Primiero et al., 2017))**.** *An* (un)$\texttt{SecureND}^{sim*}$ *judgement* $\Gamma^I \vdash \phi^I$ *is satisfied if there is a derivation* D *and a branch* $D' \subseteq D$ *with a final step terminating with such a judgement.*

Definition 12 (Validity, (Primiero et al., 2017))**.** *An* (un)$\texttt{SecureND}^{sim*}$ *judgement* $\Gamma^{\mathcal{S}} \vdash \phi^I$ *is valid if there is a derivation* D *and for all branches* $D' \subseteq D$ *and for all agents* $I \in \mathcal{S}$*, if there is a final step terminating with such a judgement.*

Example 3. *A derivation of message passing assuming* $I \leq J$*:*

$$\dfrac{\dfrac{\Gamma^I \vdash Read(\phi^J \vee \psi^J) \qquad \Gamma^I \vdash \phi^I}{\Gamma^I \vdash Trust(\phi^J \vee \psi^J)}\, Trust}{\Gamma^I \vdash Write(\phi^I \vee \psi^I)}$$

This derivation illustrates a message $\phi^J \vee \psi^J$ *transmitted by an agent* J*, read by an agent* I*. This derivation holds for both* $I \in \texttt{standard(I)}$ *or* $I \in \texttt{paranoid(I)}$*.*

Example 4. *A derivation of message passing, we assume* $I \leq J$*:*

$$\dfrac{\dfrac{\Gamma^I \vdash Read(\phi^J \to \psi^J) \qquad \Gamma^I \vdash (\phi^I \wedge \neg\psi^J)}{\Gamma^I \vdash \neg Trust(\phi^I \to \psi^I)}\, Dtrust}{\Gamma^I \vdash Write(\neg(\phi^I \to \psi^I))}$$

This derivation illustrates a message $\phi^J \to \psi^J$ *written by agent* J*, read by an agent* I *higher in the hierarchy. As the previous example, this derivation holds for both* $J \in \texttt{standard(I)}$ *or* $J \in \texttt{paranoid(I)}$*.*

$$\frac{}{\Gamma^{\mathcal{I}} \vdash Read(\phi^I)}\ read$$

$$\frac{\Gamma^I \vdash Read(\phi^J) \qquad \Gamma^I \vdash \phi^J}{\Gamma^I \vdash trust(\phi^J)}\ Trust$$

$$\frac{\Gamma^I \vdash \phi^I \qquad \Gamma^I \vdash Read(\neg\phi^J)}{\Gamma^I \vdash \neg Trust(\neg\phi^J)}\ Dtrust$$

$$\frac{\Gamma^{std(J)} \vdash Read(\phi^I) \qquad \Gamma^J \vdash \phi^I}{\Gamma^J \vdash Trust(\phi^J)}\ trust\_low\_std$$

$$\frac{\Gamma^{prd(J)} \vdash Read(\phi^I) \qquad \psi^J \vdash \phi^I}{\Delta^J \vdash Trust(\neg\psi^J)}\ Mtrust\_prd \text{ With } \Delta^J \subseteq \Gamma^J \ni \psi^J \vdash \phi^I$$

$$\frac{\Gamma^J \vdash \phi^J \qquad \Gamma^{prd(J)} \vdash Read(\neg\phi^I)}{\Gamma^J \vdash \neg Trust(\neg\phi^I)}\ Dtrust\_prd$$

$$\frac{\Gamma^{std(J)} \vdash Read(\psi^I) \to \bot \qquad \Delta^J; \psi^I : profile}{\Delta^J; \psi^J \vdash \neg Trust(\phi^J)}\ Mtrust\_std, \text{ With } \Delta^J \subseteq \Gamma^J \ni \phi^J \to \neg\psi^I$$

$$\frac{\Gamma^I \vdash Trust(\phi^J)}{\Gamma^I \vdash Write(\phi^J)}\ write\_trust \qquad \frac{\Gamma^I \vdash \neg Trust(\phi^J)}{\Gamma^I \vdash Write(\neg\phi^J)}\ distrust\_elim$$

$$\frac{\Gamma^I \vdash Write(\phi^J)}{\Gamma^I \vdash \phi^I}\ Derive$$

Figure 3.3: The system (un)$\texttt{SecureND}^{sim}$: Access rules

Example 5. *A paranoid derivation of message passing, we assume $I \leq J$ and $paranoid(J)$:*

$$\frac{\psi^J \vdash \phi^I \vee \psi^I \qquad \dfrac{\Gamma^J \vdash Read(\phi^I \vee \psi^I) \qquad \Gamma^J \vdash \psi^I}{\Delta^J \vdash Trust(\neg\psi^J)}\ Mtrust\_prd}{\Delta^J \vdash Write(\neg\psi^J)}$$

This derivation illustrates a message $\phi^I \vee \psi^I$ written by agent I, read by a paranoid agent J. If $J \in \texttt{standard(I)}$, the message $\phi^I \vee \psi^I$ would have been trusted, by an application of the $Trust\_low_std$

Example 6. *A standard derivation of message passing, we assume $I \leq J$ and $standard(v_j)$:*

$$\frac{\Delta^J; \phi^I \wedge \psi^I : profile \qquad \dfrac{\Gamma^J \vdash Read(\phi^I \wedge \psi^I) \qquad \Gamma^J \vdash \neg\psi^I}{\Delta^J; \phi^I \wedge \psi^I \vdash \neg Trust(\neg\psi^J)}\ Mtrust\_std}{\Delta^J \vdash Write(\psi^J)}$$

This derivation illustrates a message $\phi^I \wedge \psi^I$ written by agent I, read by a standard agent J. If $J \in$ `paranoid(I)`, the message would have been distrusted.

3.3 Semantics

Since connectives are added, we can formulate a relational semantic based on the one provided for `(un)SecureND` in (Primiero, 2019). As in (Primiero, 2019), the relational semantics for the system is formulated in terms of an interpretation of the accessibility relations, where the meaning of the access rules expresses conditions on them. We refer to the information held by agents as local states, while we refer to global states for the states resulting from access operations. However, the current semantic will be additionally influenced by the presence of different epistemic agents that induce further conditions on the states. Conditions of access operations have the following informal meanings:

- *Read*: a local state of an agent B is accessible by a local state of agent A, if the information available at λ_i^B is issued by B at a time earlier or at most as late as the time of state λ_j^A for agent A. In (Primiero, 2019), to access a local state is also necessary that agent A is authorized to access information from B, i.e $A \leq B$. However, in this framework of the logic, reading is not constrained on the order relation: every agent is authorized to read, keeping in mind that the order relation is essential for the characterization of the different epistemic attitudes.

- *Trust*: In (Primiero, 2019) trust requires the same condition as for read, with the addition that the information available at λ_i^B is consistent with the information available at λ_j^A. The same holds for the current framework, however a further condition is necessary: trust is needed if a consistent message is transmitted upward, whether the receiver is paranoid or is standard. For the latter, trust is called also in the case the consistent message is transmitted downward.

- *Write*: the same condition as for trust, with the addition that the information available at λ_j^A becomes visible at successive states of any authorised agent (Primiero, 2019).

- *Distrust*: in (Primiero, 2019), a message that fails the consistent requirement is distrusted. In the current system, it is applied in two different situations: when both the paranoid and the standard agent receive a contradictory message coming from below, and when a paranoid agent reads an inconsistent message coming from the top.

- *MTrustStd*: The failure of the consistent requirement of a message received by a standard agent from an agent higher in the order, it is followed by the removal on the visibility condition given by write on a locally available information (Primiero, 2019).

- *MTrustPrd*: When a paranoid agent reads a consistent message coming from an agent higher in the ranking, it removes the states from which the incoming message is derivable, while adding the negation of such states.

The semantics makes explicit the temporal succession of the agent's state, whereas it was implicit for the proof theory.

As in (Primiero, 2019), we assume a denuemerable set of atomic proposition AP.

Definition 13 (Relational model)**.**

$$\mathcal{M} := \langle \mathcal{A}, \leq, \Lambda_{A\in\mathcal{A}}, \preceq, \alpha_n^{\cdot}\omega_1, U^{\Lambda_{I,\ldots,J}}, v\rangle$$

such that

- $\mathcal{A}$:= $\{standard(I), paranoid(I)\}$ is the set of the agents, in the proof theory is the set denoted by $\mathcal{S}$.
- $\leq\subseteq \mathcal{A}\times\mathcal{A}$ is the order relation between agents. As opposed to (Primiero, 2019), it does not rule the access relation between the agents, but it represents an informal notion of the reputation and power of the agents, that triggers different behaviour depending on the epistemic attitude of the agent who is reading.
- $\Lambda_{I\in\mathcal{A}} := \Lambda_1^I, \ldots, \Lambda_n^I$ is a finite set of states for each agent $I \in \mathcal{A}$. We use the convention that λ_i^A denotes the ith local state of agent $A \in \mathcal{A}$.
- $\preceq\subseteq \Lambda_A \times \Lambda_B$ is the total temporal order relation over local states of agents. $\lambda_i^B \preceq \lambda_j^A$ express that the state held at λ_i^B is issued at the same time or earlier of the state held at λ_j^A. As in (Primiero, 2019) this relation is assumed to be reflexive, transitive and serial.
- α_n in (Primiero, 2019) denotes the the latest state of the highest ranked agent, i.e. the agent with the most authorized access over other agents. In this framework, as a consequence of the different meaning of the order relation, it identifies the latest state among the agents.
- ω_1, instead, designates the earliest state among the agents.
- $U^{\Lambda_{I,\ldots,J}} := L_I \times \cdots \times L_J$ is the Cartesian product for each agent $I \in \mathcal{A}$. As for (Primiero, 2019), we call such set an universe of states and its elements global states. For brevity of notation, in the following U^{λ_I} is denoted by U^I and $U^{\Lambda_{I,\ldots,J}}$ is denoted by $U^{\mathcal{A}}$. The maximal set $U^{\Lambda_{I,\ldots,J}}$ includes all formulas up to the latest temporal state that are satisfied in the Filter Model by Global Satisfaction, and its cardinality is denoted by 1; the minimal set U^{ω} includes only formulas up to the earliest temporal state that are satisfied in the Model by Local Satisfaction, and its cardinality is denoted by *min* (Primiero, 2019).
- $v : AP \to U^{I,\ldots,J}$ is the labelling function for atomic propositions. Intuitively $v(a^A)$ identifies the set of states in $U^{\mathcal{A}}$ where a indexed by agent A holds. A selection of states in $U^{A,\ldots,\Omega}$ might not entail an hereditary function v with respect to $\preceq$, that is,$\lambda_i^A \in v(a^A)$ and $\lambda_i^B \preceq \lambda_i^A$ is not enough to establish $\lambda_i^B \in v(a^A)$. However, The hereditary condition is always satisfied with a final selection on the model (Primiero, 2019).

Definition 14 (Local satisfaction, (Primiero, 2019)). *Given a (un)SecureND formula ϕ and a model as above, we define the satisfaction of ϕ at a specific state λ_j^A for an agent A by induction as follows:*

- $\lambda_j^A \vDash p^A$ *iff* $\lambda_j^A \in v(p^A)$
- $\lambda_j^A \vDash \top$ *for every* λ_j^A
- $\lambda_j^A \vDash \bot$ *never*
- $\lambda_j^A \vDash \phi^A \vee \psi^A$ *iff* $\exists \lambda_i^A \preceq \lambda_j^A$ *such that* $\lambda_i^A \vDash \phi^A or \lambda_i^A \vDash \psi^A$
- $\lambda_j^A \vDash \phi^A \wedge \psi^A$ *iff* $\lambda_j^A \vDash \phi^A and \lambda_j^A \vDash \psi^A$
- $\lambda_j^A \vDash \phi^A \rightarrow \psi^A$ *iff* $\exists \lambda_j^A \succeq \lambda_i^A$ *such that* $\lambda_j^A = \{Cn(\lambda_j^A \cup \phi^A)\}$ *and* $\lambda_j^A \vDash \psi^A$

An atom is satisfied at a given local state if and only if the latter belongs to the image of the labelling function of the atom. Every local state is consistent, while inconsistent states are not allowed. A disjunction is satisfied in a local state if one of the component of the disjunction is satisfied on a earlier state. A conjunction is satisfied at a local state if both the component are satisfied in that state. As in (Primiero, 2019), conjunction and disjunction are differently influenced by the temporal relation. The disjunction admits any previously held content that satisfies the relation, while the conjunction current state only where hereditary validity of formulae is preserved. An implication is satisfied at a local state if there is a successive state whose set of valid formulae is in the consequence set of the union of the current state with the antecedent of the implication, and it is satisfied the consequent of the implication. The negation connective is defined in terms of implication and $\bot$. Note that $\lambda_i^A \vDash 1$ if and only $\lambda_i^A \equiv \alpha_n$, i.e. the set of satisfied formulas has the largest cardinality if the local state of evaluation is the latest temporal state; and $\lambda_j^A \vDash min$ iff $\lambda_j^A \equiv \omega_1$, i.e. the set of satisfied formulas has the least cardinality if the state of evaluation is the earliest temporal state (Primiero, 2019).

The notion of satisfiability corresponds to validity in the local states of any given agent:

Definition 15 (Satisfiability, (Primiero, 2019)). *A formula ϕ^A is true in a model $\mathcal{M}$, denoted $\mathcal{M} \vDash \phi^A$, if and only if $\lambda_i^A \in U^{\mathcal{A}} \vDash \phi^A$ for every $\lambda_j^A \succeq \lambda_1^A \in U^A$.*

The relation of local satisfaction is monotonic, i.e. if $\lambda_i^A \in v(a^A)$ for all $\lambda_j^A \succeq \lambda_i^A$ it holds $\lambda_j^A \in v(a^A)$. Given that λ_n^A denotes the the maximally consistent set of formulas for agent A, it follows that if $\vdash \lambda_i^A$ then $\lambda_i^A \in v(a^A)$. However, a contradictory state could arise while extending a local state of a given agent to a local state of a different one. In order to preserve global monotonicity, the model requires that some local states are dismissed in view of incoming contradictory information. The dismissed states can be the one of the sender, by distrusting them, or the one of the receiver, by an operation of mistrust, in either cases it is necessary to filter these states from the model. The notion of filter model satisfies this requirement:

Definition 16 (Filter model, (Primiero, 2019)). *A filter model $\mathcal{M}'$ of $\mathcal{M}$ is a structure constructed according to Definition 13 such that $U^{\mathcal{A}} \in \mathcal{M}'$ is obtained by $U^{\mathcal{A}} \in \mathcal{M}$ by a new selection in $\Lambda_A \times \cdots \times \Lambda_\Omega$. Such selection of states and*

the addition of possibly new local states in $U^{\mathcal{A}}$ results from the Global Satisfaction Relation in Definition 17. Filter models of a given class are defined as those which select the same subset from $U^{\mathcal{A}} \in \mathcal{M}$.

A global satisfaction is defined across distinct agents in $\mathcal{A}$ for the access rules:

Definition 17 (Global satisfaction)**.** *Given a formula ϕ, a filter model as by Definition 16 above and the notion of local satisfaction it inherits, we define satisfaction of ϕ at a state λ_i^A for an agent A in a universe $U^{\mathcal{A}}$ by induction as follows:*

- $\lambda_i^A \in U^{\mathcal{A}} \vDash Read(\phi^B)$ *iff*
  1. $\exists \lambda_i^B$ *s.t* $\lambda_i^B \preceq \lambda_i^A$ *and*
  2. $\lambda_i^B \vDash \phi^B$
- $\lambda_i^A \in U^{\mathcal{A}} \vDash Trust(\phi^B)$ *iff*
  1. $A \leq B$ *or* $B \leq standard(A)$
  2. $\exists \lambda_i^B$ *s.t* $\lambda_i^B \preceq \lambda_i^A$ *and*
  3. $\lambda_i^B \vDash \phi^B$
  4. $\exists \lambda_j^A \in U^{\mathcal{A}}$ *s.t* $\lambda_i^A \preceq \lambda_j^A$ *and*
  5. $\lambda_j^A = \{Cn(\lambda^A \cup \phi^B)\}$
- $\lambda_i^A \in U^{\mathcal{A}} \vDash Write(\phi^B)$ *iff*
  1. $\exists \lambda_i^B$ *s.t* $\lambda_i^B \preceq \lambda_i^A$ *and*
  2. $\lambda_i^B \vDash \phi^B$
  3. $\exists \lambda_j^A \in U^{\mathcal{A}}$ *s.t* $\lambda_i^A \preceq \lambda_j^A$ *and*
  4. $\lambda_j^A = \{Cn(\lambda^A \cup \phi^B)\}$ *and*
  5. $\exists \lambda_k^A \in U^{\mathcal{A}}$ *s.t* $\lambda_j^A \preceq \lambda_k^A$ *and*
  6. $\lambda_k^A \vDash \phi^A$
- $\lambda_i^A \in U^{\mathcal{A}} \vDash Dtrust(\phi^B)$ *iff*
  1. $A \leq B$ *or* $B \leq paranoid(A)$
  2. $\exists \lambda_i^B$ *s.t* $\lambda_i^B \preceq \lambda_i^A$ *and*
  3. $\lambda_i^B \vDash \phi^B$ *and*
  4. $\exists \lambda_j^A \in U^V$ *s.t* $\lambda_i^A \preceq \lambda_j^B$ *and*
  5. $\lambda_j^A = \{Cn(\lambda_j^A \cup \neg\phi^B)\}$
- $\lambda_i^B \in U^V \vDash Mtrust\_std(\phi^B)$ *iff*
  1. $\exists \lambda_h^B$ *s.t* $\lambda_h^B \vDash \phi^B$ *and*
  2. $A \leq standard(B)$ *and*
  3. $\exists \lambda_i^A$ *s.t* $\lambda_i^B \succeq \lambda_i^A$ *and*
  4. $\lambda_i^A \vDash \neg\phi^A$

5. $\exists \lambda_j^B \in U^{\mathcal{A}}$ *s.t* $\lambda_i^B \preceq \lambda_j^B$ *and*
6. $\lambda_j^B = \{Cn(\lambda_i^B \setminus \{\phi^B\})\}$

- $\lambda_i^B \in U^{\mathcal{A}} \vDash Mtrust\_prd(\neg\psi^B)$ *iff*

1. $\exists \lambda_e^B$ *s.t* $\lambda_e^B \vDash \psi^B$ *and*
2. $A \leq paranoid(B)$ *and*
3. $\exists \lambda_i^A$ *s.t* $\lambda_i^B \succeq \lambda_i^A$ *and*
4. $\lambda_i^A \vDash \phi^A$ *s.t* $\psi^B \rightarrow \phi^A$ *where* $\lambda_f^B \vDash \psi^B \rightarrow \phi^A$ *iff* $\exists \lambda_g^B \succeq \lambda_f^B$ *such that* $\lambda_g^B = \{Cn(\lambda_g^B \cup \psi^B)\}$ *and* $\lambda_g^B \vDash \phi^A$
5. $\exists \lambda_j^B \in U^{\mathcal{A}}$ *s.t* $\lambda_i^B \preceq \lambda_j^B$ *and*
6. $\lambda_j^B = \{Cn((\lambda_i^B \setminus \{\psi^B\}) \cup \{\neg\psi^B\})\}$

According to these definitions, a local state λ_i^A for an agent A in a given universe $U^{\mathcal{A}}$:

- can read ϕ^B if it issued by B at a time earlier than the state of A;
- can trust ϕ^B if A is higher in the ranking than B or A is a standard agent lower in the ranking than B, the conditions of reading are satisfied and ϕ^B is consistent with at least one of the posterior states of A;
- can write ϕ^B if it can read it, trust it and relabel it so that it is satisfied in at least one posterior state;
- distrusts ϕ^B if A is a standard agent higher in the ranking than B, and ϕ^B is in contradiction with a previous state of A. Moreover, ϕ^B is distrusted if the contradictory message is received by a paranoid agent, whatever it is the position of the sender in the hierarchy;
- it can mistrust ϕ^B in two different forms:

1. a standard agent A receiving a contradictory message ϕ^B from the authority causes the removal of her own states contradicting with ϕ^B.
2. if ϕ^B is a consistent message coming from above, received by $paranoid(A)$, A will remove the state from which ϕ^B is derivable, while adding the negation of such states.

Definition 18 (Validity)**.** *A formula ϕ^A is valid in a class of filter models if and only if $\lambda_i^A \in U^{\mathcal{A}} \vDash \phi^A$ for every $\lambda_i^A \succeq \omega_A$ and every $U^{\mathcal{A}}$ in that class.*

3.4 Meta-Theory

This section presents soundness and completeness results. Before introducing them, it is necessary to investigate some proprieties related to filter models $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$.

Theorem 3. *For any two agents I, J, it never exists a filter model $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$ such that $I \leq paranoid(J)$*

Proof. Consider $I, paranoid(J) \in \mathcal{M}'$ and $\lambda_i^J \vDash Read(\phi^I)$, there are two possibilities:

1. if $\lambda_i^J \vDash Read(\phi^I) \rightarrow \bot$, then $\lambda_j^J \in U^J \vDash Distrust(\phi^I)$ and there is no filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$ or ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \neg\phi$;

2. if $\lambda_i^J \vDash Read(\phi^I)$ and $\lambda_i^J \vDash \phi^I$ then $\lambda_j^J \in U^J \vDash Mtrust\_prd(\neg\psi^J)$ for every $\psi^J \rightarrow \phi^I$. So there is no filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$ or ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \neg\phi$.

   Consider now $\lambda_i^I \vDash Read(\phi^J)$, there are two possibilities:

3. if $\lambda_i^I \vDash Read(\phi^J) \rightarrow \bot$, then in $\mathcal{M}$ it holds $\lambda_i^J \vDash Distrust(\phi^I)$, and there is no filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$ or ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \neg\phi$;

4. if $\lambda_i^I \vDash Read(\phi^J)$ and $\lambda_i^I \vDash \phi^J$), then $\lambda_j^I \in U^I \vDash Trust(\phi^J)$. However, the accessibility relation is reflexive; there will be a successive state $\lambda_k^J \vDash Read(\phi^I)$ and $\lambda_k^J \vDash \phi^I$ and then $\lambda_l^J \in U^J \vDash Mtrust\_prd(\neg\psi^J)$ for every $\psi^J \rightarrow \phi^I$. So there is no filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$ or ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \neg\phi$.

□

Theorem 3 establishes that it never exists a consistent filter Model $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$ if the information exchange involves a paranoid agent lower in the hierarchy. Indeed, *Mtrust_prd* will prevent the formation of a consistent model even if a consistent message is received by a paranoid agent.

Theorem 4. *For any two agents I, J, it always exists a filter model $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$ if $I \leq standard(J)$*

Proof. Consider $I \leq standard(J)$ and $\lambda_i^I \vdash Read(\phi^J)$, there are two possibilities:

1. if $\lambda_i^I \vDash Read(\phi^J)$ and $\lambda_i^I \vDash \phi^J$, then $\lambda_j^I \in U^I \vDash Trust(\phi^J)$. So there is a filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$;

2. if $\lambda_i^I \vDash Read(\phi^J) \rightarrow \bot$, then $\lambda_j^I \in U^I \vDash Distrust(\phi^J)$. However, the accessibility relation is reflexive: there will be a successive state $\lambda_k^J \vDash Read(\phi^I) \rightarrow \bot$ and then $\lambda_l^J \in U^J \vDash Mtrust\_std(\neg\phi^J)$. So there is a filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$.

   Consider now $\lambda_i^J \vDash Read(\phi^I)$, there are two possibilities:

3. if $\lambda_i^J \vDash Read(\phi^I)$ and $\lambda_i^J \vDash \phi^I$, then $\lambda_j^J \in U^J \vDash Trust(\phi^I)$. So there is a filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$.

4. $\lambda_i^J \vDash Read(\phi^I) \rightarrow \bot$. $\lambda_j^J \in U^J \vDash Mtrust\_std(\neg\phi^J)$. So there is a filter model ${\mathcal{M}'}^{\Lambda_{I,J}} \vDash \phi$.

□

Theorem 4 establishes that a filter model $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$ always exists when the accessibility relation involves a standard agent lower in the dominance relation. Indeed *Mtrust_std* has the benefit, as opposed to *Mtrust_prd*, of always allowing the formation of a consistent model, even if the message received by a standard agent is not consistent.

Corollary 1. *It exists a filter model $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$ iff $I \leq standard(J)$*

Proof. This follows directly from Theorem 4 which showed that for any $I \leq standard(J)$ it always exists a filter model $\mathcal{M}'$ for $U^{\Lambda_{I,J}}$, and from Theorem 3 which excluded all the others possibilities. □

Theorem 3 and theorem 4 provide an interesting characterization of the standard and the paranoid's behaviour. The paranoid attitude, as it should be expected, prevents the formation of a consistent filter model $\mathcal{M}'$, meaning that she never aligns with the position of the authority. Indeed, the presence of a paranoid agent, not placed in the top of the ranking, always causes a polarization of the formulae held by the agents: from one side we will have agents holding for example ϕ, from the other agents holding for example $\neg\phi$. Therefore, a valid formula cannot be obtained if the transmission involves a paranoid agent reading information coming from above, causing transmission that never reaches consensus. In the next chapter, this propriety of the paranoid agent will be tested experimentally by considering also other factors, such as the network topology. Opposed to the paranoid behaviour, the standard attitude has the property, thanks to the operation of mistrust, to conform always on what the authority has transmitted to her. This means that an information transmission involving only standard agents never leads to polarization and therefore consensus is always reached. An interesting aspect, based on these considerations, is that the paranoid agent does not prevent the formation of a consistent model only in the case she occupies herself the role of the authority.

Theorem 5 (Soundness)**.** *If a judgement $\Gamma^I \vdash \psi^J$ is provable in* `(un)SecureND`*, then ψ^J is true in all filter models of a given class iff ψ^I holds as well in the same class.*

Proof. In a filter model $\mathcal{M}'$ holds both ψ^J and ψ^I in the same class iff $I \leq standard(J)$ as it has been demonstrated by lemma 1. This condition guarantees that an information passing operation among a generic agent and a standard agent lower in the hierarchy always generates a consistent state which is satisfied in the filter model $\mathcal{M}'$. Therefore, if the judgment $\Gamma^I \vdash \psi^I$ is derivable then $\Gamma^I \vdash Trust(\psi^J)$ or $\Gamma^I \vdash Trust\_low\_std(\psi^J)$ must occur, then $\lambda_j^I = \{Cn(\lambda^I \cup \psi^J)\}$ and for every λ_k^I s.t $\lambda_j^A \preceq \lambda_k^A$, $\lambda_k^I \vDash \psi^I$. Finally, ψ^I holds in the same class of filter models that satisfies ψ^J. □

Theorem 6 (Completeness)**.** *If $\mathcal{M}' \vDash \phi^I$, then there is a branch of a derivation tree in* (un)`SecureND`$^{\mathtt{sim*}}$ *terminating in $\Gamma^J \vdash \phi^I$.*

Proof. We assume $\Gamma^J \nvdash \phi^I$. If the judgment $\Gamma^J \nvdash \phi^I$ is derivable then $\Gamma^J \nvdash Trust(\psi^I)$ must occur, then $\lambda_j^J = \{Cn(\lambda^J \cup \neg\phi^I)\}$ and for every λ_k^J s.t $\lambda_j^J \preceq \lambda_k^J$, $\lambda_k^J \nvdash \phi^I$. □

Chapter 4

Experimental Setting

This chapter introduces the design and the implementation of (un)SecureND^{sim*}. The main goal is to investigate consensus-reaching transmission in the presence of the paranoid attitude. Besides that, the intention is to analyze the paranoid behaviour's influence on the other agents and to propose approaches that will limit its impact on the system. As in (Primiero et al., 2017), the formal rules are implemented in NetLogo, a programming language well suited for modelling multi-agents system.

4.1 Design

The rules of the logic are implemented in different graph's topologies, where exchanges of information between the nodes represents derivations of the logic. Therefore, as in (Primiero et al., 2017), a preliminary to implement the system is the following lemma, which establishes a correspondence with a derivation D and a graph G:

Lemma 5. *[Primiero et al. (2017)] For each* (un)SecureND^{sim*} *derivation D with a valid formula $\Gamma^{\mathcal{S}} \vdash \phi^{\mathcal{S}}$, there is a graph G that is unanimously labelled by ϕ.*

Proof. The proof requires to construct a graph G with a node for each distinct $I \in \mathcal{S}$ occurring in D and an edge for each judgement instantiating one or more rules with two distinct nodes on each side of the derivability sign. Starting from the node occurring at the highest position of D validating ϕ, by application of one or more sequences of rules the conclusion in such branch of D represents a new node in G labelled by ϕ. If all branches of D terminate with a formula validating ϕ, as by assumption and according to Definition 4, then all nodes in G will be labelled by ϕ. □

Hereafter we define basic concepts needed to develop the implementation:

Definition 19 (Graph, (Primiero et al., 2017))**.** *A network is an undirected graph $G = (V, E)$, with a set $V = \{v_i, \ldots, v_n\}$ of vertices representing our agents and a set $E = \{e_{(i,j)}, \ldots, e_{(n,m)}\}$ of edges, representing transmissions among them.*

Definition 20 (Labelling, (Primiero et al., 2017))**.** *Each vertex $v_i \in V$ can be labelled by formulas as follows:*

- *$v_i(p)$ denotes a vertex labelled by an atomic formulas and expresses an agent i knowing p;*
- *$v_j(\neg p)$ denotes a vertex labelled by the negation of an atomic formula and expresses agent j knowing $\neg p$;*
- *$v_k()$ denoted a vertex with no label and expresses an agent k who does not hold any knowledge yet.*

As for the logic we maintain that a vertex v_i can be either standard or paranoid. The transmission between two nodes is expressed by an edge representing a *transmission channel*. A channel is denoted for example by $e(v_i(p), v_j())$; in this particular case i transmits p to j, where j represents a node yet not labelled. Different kinds of transmission are allowed, but they need distinct procedures to establish whether the information is accepted or not, depending on the epistemic attitude of the agent receiving it. Another factor that influences the transmission is the order relation $\leq$, which is defined in the logic as a total order over $\mathcal{S} \times \mathcal{S}$. However, in this context it becomes total or partial depending on the topology of the network of interest. In particular, two different network topologies are considered:

- In a *total network* each vertex has an edge connected to any other and equal ranking is assigned to all the agents, hence the order is total;
- In a *scale-free network* the creation of edges follows a power-law degree distribution; newly added nodes tend to prefer vertices with a high number of links. The Scale-free network is shaped, as in (Primiero et al., 2017), following the Barabasi-Albert method to establish edges:

  > Initialised by $m = 3$ nodes, each node with 0 neighbours is asked to create an edge with a vertex in the network; for each new vertex v_j without neighbours, v_j is connected to up to $n < m$ existing vertices with a probability $\mathbf{p}(v_j)$ defined by the following expression:
  >
  > $$\mathbf{p}(v_j) = \frac{k_{v_j}}{\sum_{v_i} k_{v_i}}$$
  >
  > where k_{v_j} is the number of neighbours of agent v_j and the sum is made over all pre-existing nodes v_i. Newly added nodes tend to prefer nodes that already have a higher number of links. The ranking in this case is given to each node by a simple function $\frac{1}{|edges|}$.

  Particular attention will be given to this topology, since interesting real-world networks follow the scale-free model.

Differently from the logic, only atomic formulae p and $\neg p$ are considered. Every information-exchange between the nodes of the graph starts with a randomly seeded information p, which spreads across the network according to the algorithm `Transmission`$*$ presented in Figure 4.1. The transmission of the information mimics the one of the logic: different operations are called by the standard and paranoid's behavior depending on their position in the hierarchy. In summary, a standard node

PROCEDURE Transmission$*(G)$
$G := (V, E)$

FOR $e(v_i(\phi), v_j()) \in G$ (the same holds **for** $v_j(p)$)
IF (`ranking`$(v_i(\phi)) <$ `ranking`$(v_j()))$ AND $(v_j() \in$ `paranoid`$)$
THEN `Mtrust_prd`$(e(v_i(\phi), v_j()))$
ELSEIF $(v_j() \in$ `standard`$)$ OR $(v_j() \in$ `paranoid`$)$ AND (`ranking`$(v_j()) \leq$ `ranking`$(v_i(\phi)))$
THEN `Trust`$(e(v_i(\phi), v_j()))$
ENDIFELSE
ENDFOR

FOR $e(v_i(\phi), v_j(\neg\phi)) \in G$
IF (`ranking`$(v_i(\phi)) <$ `ranking`$(v_j(\neg\phi)))$ AND $(v_j(\phi) \in$ `standard`$)$
THEN `Mtrust_std`$(e(v_i(\phi), v_j(\neg\phi)))$
ELSEIF $(v_j(\neg\phi) \in$ `standard`$)$ OR $((v_j(\neg\phi) \in$ `paranoid`$)$ AND (`ranking`$(v_j(\neg\phi)) \leq$ `ranking`$(v_i(\phi)))$
THEN `DTrust`$(e(v_i(\phi), v_j(\neg\phi)))$
ENDIFELSE
ENDFOR

FOR $(e(v_i(\phi), v_j(), v_k(\neg\phi)) \in G)$ AND ((`ranking`$(v_i) \leq$ `ranking`$(v_k) <$ `ranking`$(v_j))$
`SolveConflict`$(e(v_i(\phi), v_j(), v_k(\neg\phi)))$
ENDFOR

ENDPROCEDURE

Figure 4.1: Algorithm for Information Transmission

label itself with p if a node linked to it holds p and p is consistent in its context, i.e it holds p or it is not labeled yet. This procedure is executed by calling the `Trust` routine. Moreover, a standard vertex will accept p even if it is not consistent, i.e. it holds $\neg p$, in the case she is lower in the hierarchy than the node sending the information. This last transmission mimics the derivation rule `Mtrust_std` and it is called by the `Mtrust_std` routine. Symmetrically, if the receiver is paranoid and it receives information from above, it calls the `Mtrust_prd` routine and the node is added with an opposite label. When both paranoid and standard attitudes are simulated by vertices up in the ranking, information will be trusted if consistent, i.e it will be accepted, and will be distrusted if it is not consistent, i.e it will reject it by calling the `Dtrust` routine.

Taking into account data on the presence in the population of conspiracy theorists (Oliver & Wood, 2014), we consider three fixed distributions of the epistemic attitudes across the networks. Hence, we define three configurations of networks:

1. *balanced*: in this type of network, the proportion between paranoid and standard nodes is set at 50%. This distribution will be useful to analyze clearly the effects of the paranoid behaviours in the context of information transmission, and will give hints in order to find conditions to limit their effects;

2. *representational*: in this type of network, the proportion of paranoid nodes is set at 12%. This percentage represents people in the population of the U.S who endorse at least three conspiracy theories (Oliver & Wood, 2014). This distribution will be more suiting to model real case scenario;

3. *overly*: in this type of network we set the percentage of paranoid agents to the minimum needed to increase the chances to obtain consensus reaching transmission.

PROCEDURE `Trust`$(e(v_i(\phi), v_j()))$
$G := G \cup (v_j(\phi))$
ENDPROCEDURE

PROCEDURE `Dtrust`$(e(v_i(\phi), v_j(\neg\phi)))$
$G := G \cup (v_j(\neg\phi))$
ENDPROCEDURE

Figure 4.2: Algorithm for Trust and Distrust

PROCEDURE `MTrust_prd`$(e(v_i(\phi), v_j()))$
$G := G \cup (v_j(\neg\phi))$
ENDPROCEDURE

PROCEDURE `MTrust_std`$(e(v_i(\phi), v_j(\neg\phi)))$
$G := G \cup (v_j(\phi))$
ENDPROCEDURE

Figure 4.3: Algorithm for Mistrust

The sub-routines `Trust` and `DTrust` are illustrated in Figure 4.2. If `Trust` is called, the node is relabelled with the sender's label; instead, if `DTrust` is called, the node appends a label opposite to the one of the sender. `MTrust_std` and `Mtrust_prd` are illustrated in Figure 4.3. `Mtrust_std` is called when a standard node is linked with another higher in the ranking and with an opposite label. Therefore, by mimicking the logic, it mistrusts its current information, and it is relabelled with the sender's label. Symmetrically, if a paranoid vertex is linked with a higher node than itself in the ranking, it mistrusts its current information in order to be labelled with an opposite label of the sender's one.

Therefore, this four subroutines try to capture the main aspects of the formal system. In particular, we maintain that an agent higher in the ranking is more likely to keep her current information by trusting consistent information and by distrusting inconsistent information. Instead, agents ranked lower are more prone to change their informational state. Indeed, a standard agent accepts every information coming from above even if it is not consistent, while the paranoid agent, in order to not being aligned with the authority, will mistrust its current consistent information.

In the case a standard or a paranoid agent are linked with two nodes higher then them in the ranking and sending contradictory information, the routine `SolveConflict` is called. It lets the node chooses one of the two contradictory formulae depending on its epistemic characterization. In the occurrence of a paranoid vertex, it chooses the formula of the agent lower in the ranking. Instead, a standard one chooses the formula of the agent higher in the ranking. In this way, we maintain the principle that guided the different epistemic characterization of the logic: the paranoid attitude is more prone to trust agents lower in the hierarchy, while the standard one always trusts the authority. An equivalent of `SolveConflict` is not present in the logic, the necessity of it arises when the transmission of the information occurs at once between more than two agents as is the case in the simulation.

PROCEDURE `SolveConflict`$(e(v_i(\phi), v_j(), v_k(\neg\phi)))$
IF $v_j \in$ `standard` AND `ranking`$v_i(\phi) \leq$ `ranking`$v_k(\neg\phi)$
$G := G \cup (v_j(\phi))$
ELSEIF $v_j \in$ `paranoid` AND `ranking`$v_i(\phi) \leq$ `ranking`$v_k(\neg\phi)$
$G := G \cup (v_j(\neg\phi))$
ENDIFELSE
ENDPROCEDURE

Figure 4.4: Algorithm for Conflict Resolution

4.2 Code

In this section we present and comment the main passages of the NetLogo code developed to simulate the information transmission between standard and paranoid agents. The full code is available at https://github.com/gprimiero/paranoid.

```
turtles-own

[
 ranking
 paranoid?
 standard?
 p?
 notp?
 solveconf
 triangle?
 betweenness-centrality_notp
 betweenness-centrality_p
 discover_p?
 discover_notp?
]
```

Figure 4.5: Nodes' variable

`Turtles-own` (figure 4.5) are the variable assigned to the nodes of the graph. More precisely, we characterize the nodes by:

- a value that expresses their ranking; in the implementation, "being lower in the ranking" i.e. $n \leq m$, expresses the condition that an agent n is in higher or equal hierarchical position than agent m;
- by being paranoid or standard,
- by holding p or $\neg p$ and
- by being the nodes in which the initial information is seeded, reported by the variables `discover_p?` and `discover_notp?`.

```
to setup

  clear-all
  set-default-shape turtles "circle"

  if (_network_type = "total")
  [
   create-turtles _nodes
   [
    initializeTurtle
    set ranking 0
```

```
    ]
     ask turtles [ create-links-with other turtles ]
     layout-circle turtles 13
    ]

    if (_network_type = "small-world")
    [
     create-turtles 3 [ initializeTurtle ]
     ask turtle 0 [ create-link-with one-of other turtles ]
     ask one-of turtles with [count link-neighbors = 0] [ create-link-with one-of other turtles ]
     while [count turtles < (_nodes)]
     [
      create-turtles 1
      [
       initializeTurtle
       create-link-with find-partner
      ]
     ]
      ask turtles [ set ranking 1 / count link-neighbors ]
      ask turtles [ set size 2 - ranking ]
      layout-radial turtles links max-one-of turtles [count link-neighbors]
    ]

    set change-count 0
    epistemic_attitudes
    reset-ticks

  end
```

Figure 4.6: Setup

`To Setup` is the procedure that shapes the different networks. For every network, the maximum number of nodes is set at 300. In figure 4.6 are presented the subroutines to initiate the different network topologies taken into considerations:

- a *total network* is shaped making sure that each node is linked with every other node;
- a *scale-free network* follows the power rule explained in section 4.1 to create links with the other nodes;

The instructions for the formation of the different networks are a slightly modified version of the one used in the simulation of (Primiero et al., 2017), available at https://github.com/gprimiero/securendsim. In addition to this, the procedure calls the routine `epistemic_attitudes` presented in figure 4.7

```
to epistemic_attitudes

  let n_of_paranoid (proportion_paranoid * _nodes) / 100
  repeat n_of_paranoid
  [
   ask one-of turtles with [triangle? = 0]
   [
    set shape "triangle"
    set paranoid? true
    set standard? false
    set triangle? true
   ]
  ]

  ask turtles with [ shape = "circle"]
  [
   set standard? true
   set paranoid? false
```

```
  ]

end
```

Figure 4.7: Epistemic attitudes

When `epistemic_attitudes` is called the nodes are differentiated by paranoid and standard nodes according to the proportion chosen. To distinguish them visually, two different shapes are set: standard nodes are circle, while paranoid nodes are triangles.

```
to discovery

  if discovery_type = "paranoid_random"
  [
    ask one-of turtles with [paranoid? = true]
    [
     set color red
     set p? false
     set notp? true
     set discover_notp? true
    ]
  ]

  if discovery_type = "standard_random"
  [
    ask one-of turtles with [standard? = true]
    [
     set color green
     set p? true
     set notp? false
     set discover_p? true
    ]
  ]

  if discovery_type = "standard_min"
  [
   ifelse count turtles with-min [ranking] with [standard? = true ] = 0
    [minimize]
    [
     ask one-of turtles with-min [ranking] with [standard? = true]
     [
      set color green
      set p? true
      set notp? false
      set discover_p? true
     ]
    ]
   ]

  if discovery_type = "paranoid_min"
  [
   ifelse count turtles with-min [ranking] with [paranoid? = true ] = 0
    [minimize]
    [
     ask one-of turtles with-min [ranking] with [paranoid? = true]
     [
      set color red
      set p? false
      set notp? true
      set discover_notp? true
     ]
    ]
   ]

 if discovery_type = "standard_max"
```

```
  [
  ifelse count turtles with-max [ranking] with [standard? = true ] = 0
   [maximize]
   [ask one-of turtles with-max [ranking] with [ standard? = true]
    [
     set color green
     set p? true
     set notp? false
     set discover_p? true
    ]
   ]
  ]

  if discovery_type = "paranoid_max"
   [
    ifelse count turtles with-max [ranking] with [paranoid? = true ] = 0
    [maximize]
    [
     ask one-of turtles with-max [ranking] with [paranoid? = true]
     [
      set color red
      set p? false
      set notp? true
      set discover_notp? true
     ]
    ]
   ]
```

Figure 4.8: Discovery types

`To discovery` is the procedure that establishes in which kind of nodes is seeded the initial information. If we choose the discovery type `standard_random`, the information p is given to a random standard node. Instead, if we choose the discovery type `standard_min`, the information p is seeded in the standard node with the lowest ranking. Finally, if `standard_max` is chosen, the information p is offered to the standard node with the highest ranking. The same rationale holds for `paranoid_random`, `paranoid_min` and `paranoid_max` respectively; however, the seeded information in paranoid nodes is $\neg p$.

```
if discovery_type = "contradictory_random"
  [
   ask one-of turtles with [paranoid? = true]
   [
    set color red
    set p? false
    set notp? true
    set discover_notp? true
   ]
    ask one-of turtles with [standard? = true]
    [
     set color green
     set p? true
     set notp? false
     set discover_p? true
    ]
  ]

  if discovery_type = "contradictory_std-min_prd-max"
   [
    ifelse count turtles with-min [ranking] with [standard? = true ] = 0
    [minimize]
     [
      ask one-of turtles with-min [ranking] with [standard? = true]
       [
```

```
      set color green
      set p? true
      set notp? false
      set discover_p? true
     ]
    ]
   ifelse count turtles with-max [ranking] with [paranoid? = true ] = 0
   [maximize]
    [
     ask one-of turtles with-max [ranking] with [paranoid? = true]
     [
      set color red
      set p? false
      set notp? true
      set discover_notp? true
     ]
    ]
   ]

  if discovery_type = "contradictory_std-max_prd-min"
   [
    ifelse count turtles with-min [ranking] with [paranoid? = true ] = 0
    [minimize]
     [
      ask one-of turtles with-min [ranking] with [paranoid? = true]
      [
       set color red
       set p? false
       set notp? true
       set discover_notp? true
      ]
     ]
    ifelse count turtles with-max [ranking] with [standard? = true ] = 0
    [maximize]
     [
      ask one-of turtles with-max [ranking] with [standard? = true]
      [
       set color green
       set p? true
       set notp? false
       set discover_p? true
      ]
     ]
    ]

  end
```

Figure 4.9: Contradictory discoveries

Contradictory discoveries are available too (see figure 4.9). This means that two atoms of initial information are seeded in the network: p to a standard agent and $\neg p$ to a paranoid one. Three possibilities are considered:

- if `contradictory_random` is chosen, a random standard agent is labelled with the formula p, and a random paranoid vertex is labelled with $\neg p$;
- if `contradictory_std-min_prd-max` is selected, a standard node with the minimum ranking is labelled with p, and $\neg p$ is assumed by the the paranoid nodes with the maximum ranking;
- if `contradictory_std-max_prd-min` is picked, $\neg p$ is given to the paranoid node with the minimum ranking and p to the the standard nodes with the maximum ranking

The discoveries concerning maximum and minimum of the ranking employ the routines minimize and maximize, which are presented in figure 4.10 and figure 4.11.

```
to minimize

  if discovery_type = "standard_min" or discovery_type = "contradictory_std-min_prd-max"
  [
   ask one-of turtles with-min [ranking]
   [
    set shape "circle"
    set standard? true
    set paranoid? false
    set color green
    set p? true
    set notp? false
    set discover_p? true
    ask one-of other turtles with [standard? = true]
    [
     set shape "triangle"
     set paranoid? true
     set standard? false
    ]
   ]
  ]

  if discovery_type = "paranoid_min" or discovery_type = "contradictory_std-max_prd-min"
  [
   ask one-of turtles with-min [ranking]
   [
    set shape "triangle"
    set paranoid? true
    set standard? false
    set color red
    set p? false
    set notp? true
    set discover_notp? true
    ask one-of other turtles with [paranoid? = true]
    [
     set shape "circle"
     set standard? true
     set paranoid? false
    ]
   ]
  ]

end
```

Figure 4.10: Minimize

`To minimize` is called when a discovery concerning the minimum of the ranking is called, and no node of the epistemic attitude considered is the minimum in the current configuration of the network. Therefore, `To minimize` allows to ask one node with the minimum ranking to change its epistemic characterization, depending on the epistemic attitude in which we want to seed the initial information. Additionally, in order to keep unaltered the proportion between paranoid and standard nodes, it asks another node to became paranoid if the "minimized" node is standard, or standard if the "minimized" node is paranoid. In this way, we guarantee to have a minimally ranked node of the chosen attitude as seed, while the overall proportion is kept untouched.

```
to maximize

  if discovery_type = "standard_max" or discovery_type = "contradictory_std-max_prd-min"
  [
   ask one-of turtles with-max [ranking]
   [
    set shape "circle"
    set standard? true
    set paranoid? false
    set color green
    set p? true
    set notp? false
    set discover_p? true
    ask one-of other turtles with [standard? = true]
    [
     set shape "triangle"
     set paranoid? true
     set standard? false
    ]
   ]
  ]

  if discovery_type = "paranoid_max" or discovery_type = "contradictory_std-min_prd-max"
  [
   ask one-of turtles with-max [ranking]
   [
    set shape "triangle"
    set paranoid? true
    set standard? false
    set color red
    set p? false
    set notp? true
    set discover_notp? true
    ask one-of other turtles with [paranoid? = true]
    [
     set shape "circle"
     set standard? true
     set paranoid? false
    ]
   ]
  ]

end
```

Figure 4.11: Maximize

Similarly, `to maximize` is called if the discovery chosen involves a paranoid or standard node and there is not any such node with the maximum ranking. As `to minimize`, when it is called, another node becomes paranoid or standard depending on the procedure called.

```
to go

  transmission
  solveconflict
  tick
  if ticks mod stability-factor = 0
  [if change-count < 1 [stop] set change-count 0]
  assign-betweennes-centrality

end
```

Figure 4.12: Go procedure

`to go` is the core of the code: when called, the seeded information starts to spread through the network following the rules established by the routine `transmission` and the procedure `solveconflict`. To stop it, we implemented a ploy inspired by the model available at http://bit.ly/3aq4c4T: `go` stops when the `change-count` is less than 1 for a certain amount of ticks (a tick represents one execution of a procedure), meaning that the system is stable and there are no more changes. In what follows, we look specifically to the procedures called by `go`.

```
to transmission

  ask turtles with [p? = true]
  [
    if any? link-neighbors with [(color = blue) and (standard? = true) and (solveconf = 0)]
    [
      ask one-of link-neighbors with [(color = blue) and (standard? = true) and (solveconf = 0)]
      [trust_p]
    ]

    if any? link-neighbors with
    [
     (notp? = true) and (standard? = true) and (ranking > [ranking] of myself) and (solveconf =
         0)
    ]
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (notp? = true) and (standard? = true) and (ranking > x) and (solveconf = 0)
      ]
      [mtrust_notp]
    ]

    if any? link-neighbors with
    [
     (color = blue) and (paranoid? = true) and (ranking <= [ranking] of myself) and (solveconf
         = 0)
    ]
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (color = blue) and (paranoid? = true) and (ranking <= x) and (solveconf = 0)
      ]
      [trust_p]
    ]

    if any? link-neighbors with
    [
     (color = blue) and (paranoid? = true) and (ranking > [ranking] of myself) and (solveconf =
         0)
    ]
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (color = blue) and (paranoid? = true) and (ranking > x) and (solveconf = 0)
      ]
      [mtrust_p]
    ]

    if any? link-neighbors with
    [
     (p? = true) and (paranoid? = true) and (ranking > [ranking] of myself) and (solveconf = 0)
    ]
```

```
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (p? = true) and (paranoid? = true) and (ranking > x) and (solveconf = 0)
      ]
      [mtrust_p]
    ]
  ]


  ask turtles with [notp? = true]
  [
    if any? link-neighbors with [(color = blue) and (standard? = true) and (solveconf = 0)]
    [
     ask one-of link-neighbors with [(color = blue) and (standard? = true) and (solveconf = 0)]
      [trust_notp]
    ]

    if any? link-neighbors with
    [
     (p? = true) and (standard? = true) and (ranking > [ranking] of myself) and (solveconf = 0)
    ]
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (p? = true) and (standard? = true) and (ranking > x) and (solveconf = 0)
      ]
      [mtrust_p]
    ]

    if any? link-neighbors with
    [
     (color = blue) and (paranoid? = true) and (ranking <= [ranking] of myself) and (solveconf
         = 0)
    ]
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (color = blue) and (paranoid? = true) and (ranking <= x) and (solveconf = 0)
      ]
      [trust_notp]
    ]

    if any? link-neighbors with
    [
     (color = blue) and (paranoid? = true) and (ranking > [ranking] of myself) and (solveconf =
         0)
    ]
    [
      let x 0
      set x [ranking] of self
      ask one-of link-neighbors with
      [
       (color = blue) and (paranoid? = true) and (ranking > x) and (solveconf = 0)
      ]
      [mtrust_notp]
    ]

    if any? link-neighbors with
    [
     (notp? = true) and (paranoid? = true) and (ranking > [ranking] of myself) and (solveconf =
         0)
    ]
    [
      let x 0
```

```
    set x [ranking] of self
    ask one-of link-neighbors with
    [
     (notp? = true) and (paranoid? = true) and (ranking > x) and (solveconf = 0)
    ]
    [mtrust_notp]
  ]
]
```

Figure 4.13: Transmission

Figure 4.13 presents the routine `transmission`. This procedure is the translation in NetLogo of the conditions expressed by the homonym algorithm (see figure 4.2). In summary, the labelled nodes, those holding p or $\neg p$, are asked if they are directly linked to other vertices. These latter nodes, depending on their epistemic characterization and on the ranking of the node sending the information, call the routines *to trust* or *to mtrust* (see figure 4.17). The procedure is divided by the sending nodes holding p and the sending nodes holding $\neg p$; while the nodes receiving the information are characterized by being standard or paranoid and by holding either p, $\neg p$ or by being not labeled yet.

```
ask turtles with [solveconf = 0]
 [
  if any? link-neighbors with [(ranking < [ranking] of myself) and (notp? = true)]
  [
   if any? link-neighbors with [(ranking < [ranking] of myself) and (p? = true)]
   [
    set solveconf solveconf + 1
    solveconflict
   ]
  ]
 ]
```

Figure 4.14: Call of solveconflict

The subroutine `to solveconflict` introduced in figure 4.14 is needed if a node is linked with two nodes higher than itself in the hierarchy, each sending contradictory information with respect to the other.

```
ask turtles with [(paranoid? = true) and (solveconf = 0)]
  [
   if notp? = true
   [
    if any? link-neighbors with [(color = blue) and (ranking < [ranking] of myself)]
    [
     if any? link-neighbors with [(p? = true) and (ranking < [ranking] of myself)]
     [
      let x one-of link-neighbors with [(color = blue) and (ranking < [ranking] of myself)]
      let currentLink in-link-from x
      ask currentLink
      [
       let mynodes [both-ends] of currentLink
       ask one-of mynodes with [(paranoid? = true) and (notp? = true) and (solveconf = 0)]
       [set solveconf solveconf + 2]
      ]
     ]
    ]
   ]
   if p? = true
   [
```

```
    if any? link-neighbors with [(color = blue) and (ranking < [ranking] of myself)]
    [
     if any? link-neighbors with [(notp? = true) and (ranking < [ranking] of myself)]
     [
      let x one-of link-neighbors with [(color = blue) and (ranking < [ranking] of myself)]
      let currentLink in-link-from x
      ask currentLink
      [
       let mynodes [both-ends] of currentLink
       ask one-of mynodes with [(paranoid? = true) and (p? = true) and (solveconf = 0)]
       [set solveconf solveconf + 2]
      ]
     ]
    ]
   ]
  ]

  ask turtles with [(solveconf = 0)]
  [
   if any? link-neighbors with [(ranking < [ranking] of myself) and (notp? = true)]
   [
    if any? link-neighbors with [(ranking < [ranking] of myself) and (p? = true)]
    [
     let m' link-neighbors with [(ranking < [ranking] of myself) and (p? = true)]
     let m link-neighbors with [(ranking < [ranking] of myself) and (notp? = true)]
     if ([ranking] of m') = ([ranking] of m) and (paranoid? = true)
     [
      set color red
      set p? false
      set notp? true
      set change-count change-count + 1
      set solveconf solveconf + 2
     ]
     if ([ranking] of m') = ([ranking] of m) and (standard? = true)
     [
      set color green
      set p? true
      set notp? false
      set change-count change-count + 1
      set solveconf solveconf + 2
     ]
    ]
   ]
  ]
```

Figure 4.15: Cases where solveconflict is not called

However, the call of `solveconflict` is avoided in two cases presented in figure 4.15:

1. when a paranoid node is linked with two agents lower than it in the ranking, among which one its labelled and one is not. In this case the paranoid node will read firstly the formula of the labelled node, then it will apply `mtrust` and it will relabel itself with an opposite formula. At this point the not yet labelled node will read the formula of the paranoid node and labelled itself with the same formula as a result of an application of `trust`. Therefore, the paranoid node will satisfy the conditions to call `solveconflict`, but since this conditions are satisfied by a direct cause of the paranoid behaviour we maintain, in this situation, that the paranoid node keeps its current label;

2. when the two vertices transmitting contradictory information have the same ranking. In this case a paranoid agent label itself with $\neg p$ while a standard one with p, i.e. they follow their own breed.

```
to solveconflict

  ask turtles with [(solveconf = 1) and (paranoid? = true)]
  [
   let m link-neighbors with-min [ranking] ask [m] of self
   [
    if notp? = true or color = blue
    [
     ask turtles with [(solveconf = 1) and (paranoid? = true)]
     [
      set color green
      set p? true
      set notp? false
      set change-count change-count + 1
      set solveconf solveconf + 1
     ]
    ]
    if p? = true
    [
     ask turtles with [(solveconf = 1) and (paranoid? = true)]
     [
      set color red
      set p? false
      set notp? true
      set change-count change-count + 1
      set solveconf solveconf + 1
     ]
    ]
   ]
  ]

  ask turtles with [(solveconf = 1) and (standard? = true)]
  [
    let m link-neighbors with-min [ranking] with [p? = true or notp? = true] ask [m] of self
   [
    if notp? = true
    [
     ask turtles with [(solveconf = 1) and (standard? = true)]
     [
      set color red
      set p? false
      set notp? true
      set change-count change-count + 1
      set solveconf solveconf + 1
     ]
    ]
    if p? = true or color = blue
    [
     ask turtles with [(solveconf = 1) and (standard? = true)]
     [
      set color green
      set p? true
      set notp? false
      set change-count change-count + 1
      set solveconf solveconf + 1
     ]
    ]
   ]
  ]

end
```

Figure 4.16: Solveconflict

In figure 4.16 is introduced the routine `solveconflict`. The procedure, as explained in figure 4.14 is called when a node is in the middle of contradictory information sent by two agents with a lower ranking than it. Therefore, it chooses p or

$\neg p$ by looking at the formula possessed by the lower agent in the ranking between the two nodes transmitting contradictory information. In the case a paranoid node is reading, it labels itself with the opposite label of the lower node in the ranking, while a standard node labels itself with the same one.

```
to mtrust_p

   set color red
   set p? false
   set notp? true
   set change-count change-count + 1

end

to mtrust_notp

   set color green
   set p? true
   set notp? false
   set change-count change-count + 1

end

to trust_p

   set color green
   set p? true
   set notp? false
   set change-count change-count + 1

end

to trust_notp

   set color red
   set p? false
   set notp? true
   set change-count change-count + 1

end
```

Figure 4.17: Trust and Mistrust

In figure 4.17 are introduced the rules for trust and mistrust, that we have formulated for the logic, translated in the NetLogo code.

```
to assign-betweennes-centrality

   ask turtles with [ notp? = true]
   [
    if any? link-neighbors with [(ranking >= [ranking] of myself) and (notp? = true)]
    [set betweenness-centrality_notp nw:betweenness-centrality]
   ]

   ask turtles with [ p? = true]
   [
    if any? link-neighbors with [(ranking >= [ranking] of myself) and (notp? = true)]
    [set betweenness-centrality_p nw:betweenness-centrality]
   ]

end
```

Figure 4.18: Betweenness centrality

`To assign-betweenness-centrality` is the last procedure called by `to go`, and it is initiated when the simulation stops. In graph theory, betweenness centrality measures the importance of a node in a network. To compute it for a node N:

> we select a pair of nodes and find all the shortest paths between those nodes. Then we compute the fraction of those shortest paths that include node N. We repeat this process for every pair of nodes in the network. We then add up the fractions we computed, and this is the betweenness centrality for node N (Golbeck, 2013)[p. 30].

A NetLogo primitive `nw:betweennes-centrality` calculates it automatically. To assign it to the nodes of interests we ask the vertices labelled with $\neg p$ or p if there are higher ranked nodes directly linked to them, which are also labelled with $\neg p$ or p respectively. When these conditions are satisfied, we assign the betweenness centrality of these nodes with the variables `betweenness-centrality_notp` and `betweenness-centrality_p` (see figure 4.5) respectively, depending on the node's label we want to calculate the betweenness of.

Chapter 5

Experimental results

Experiments are run over two distinct types of networks: scale-free and total networks. The complex topology of scale-free networks models better real case scenario, such as social networks. Instead, total networks are more representative of the logic, since the order relation is total. The experiments have been executed on a machine with 7.8 GB of memory running 64bit Windows 10. The data is collected from networks of three fixed dimensions of 50, 150 and 300 nodes and three fixed distributions of paranoids nodes (3%, 12% and 30%). In all the experiments, we seed an atom p to a standard node and its negation to a paranoid one. The result of the experiments are available at https://github.com/gprimiero/paranoid

5.1 Consensus

The two different networks configuration affect consensus. For total networks, whether the information seeded is p or is $\neg p$, consensus is reached in all the runs of the simulations. Indeed, consensus on the formula p is reached if the initial information is seeded in a standard node, and consensus on the formula $\neg p$ is reached if the initial information is seeded in a paranoid node. This is a direct consequence of Theorem 4: the ranking of the nodes is equal, therefore no operation of mistrust by paranoid nodes will occur, hence in the case of a discovery standard, no skewed information $\neg p$ will be transmitted to other nodes. The situation changes when the seeded information is contradictory: in this case information is polarized, nodes will hold p or $\neg p$ indifferently. This last case is a consequence of Theorem 3: the nodes holding $\neg p$ and the nodes holding p, independently of their epistemic characterization, keep their current labelling and distrust the other information available.

Now we analyze consensus in scale-free networks, the results are introduced in figure 5.1. Scale-free networks reach consensus rarely: only the configurations concerning a network with a dimension of 50 nodes and with the proportion of paranoid nodes set at 3% reach consensus. In this kind of configuration just one node is paranoid and the rest are standard. In particular, only 6 out of 81 of the configuration tested present some runs in which consensus is reached. This preliminary consideration is sufficient to state that, as expected, the paranoid behaviour is a hurdle for consensus reaching transmission. Let us examine the configurations in which consensus is reached for scale-free networks. We start with the the experiments in which the initial information is seeded in a standard node:

1. When the node seeded is a random standard node, only 5 runs out of 30 reach consensus.

2. When the initial information is seeded in a standard node with the minimum of the ranking, consensus is never reached. However, in the majority of such runs (25 out of 30) just the only paranoid node holds the skewed information $\neg p$. Therefore, paranoid nodes can be characterized as being particularly inflexible, since they resist in preserving their label.

3. When the information is seeded in a standard node with the maximum ranking, similarly to the random discovery, consensus is reached in 4 runs out of 30.

Let us examine runs in which consensus is reached when the initial information is seeded in a paranoid node; in this cases consensus is never reached on the formula p but on its negation, as a result of a paranoid agent transmitting it:

1. When it is seeded in a random paranoid node, consensus is never reached: the only paranoid agent in the network always holds p, while the other nodes $\neg p$. The same result is obtained when the initial information is seeded in the paranoid node with the maximum ranking. In these cases, failed consensus is a direct cause of a mistrust operation of the paranoid node, who reverses its own opinion in order to hold a formula opposed to the one of the authority. This is a direct consequence of Theorem 4, modelling the case where the transmission is between a standard node in a prominent position of the hierarchy and a paranoid node lower in the hierarchy.

2. When the initial information is seeded in the paranoid node with the minimum of the ranking, consensus is reached in all the 30 runs of the simulation. In this situation, the only paranoid node is the leading authority, therefore no operation of mistrust will impede consensus. This is proven also directly in the logic in Theorem 3: when information-transmission involves a paranoid agent above in the hierarchy and a standard one below, a model validating a formula for both agents can always be construed.

Now we analyze the cases of consensus-reaching runs where contradictory information is seeded in the network. First of all, we would like to point out that, in the case of a contradictory discovery, the influence of the standard nodes is limited, since they mistrust their information to conform with the contradictory message sent by the authority. In this kind of discovery, consensus is never reached on the information seeded in the standard node, namely p, but on $\neg p$, i.e. the information seeded in the paranoid node. The main results of contradictory discovery, concerning consensus, are presented hereafter:

1. When the nodes seeded are a random standard one with p and a random paranoid one with $\neg p$, just 1 run out of 30 reaches consensus. Specifically, all the nodes are labelled with $\neg p$, which means that the influence of the paranoid node is total: all the nodes holding p mistrusted this information at some point of the transmission.

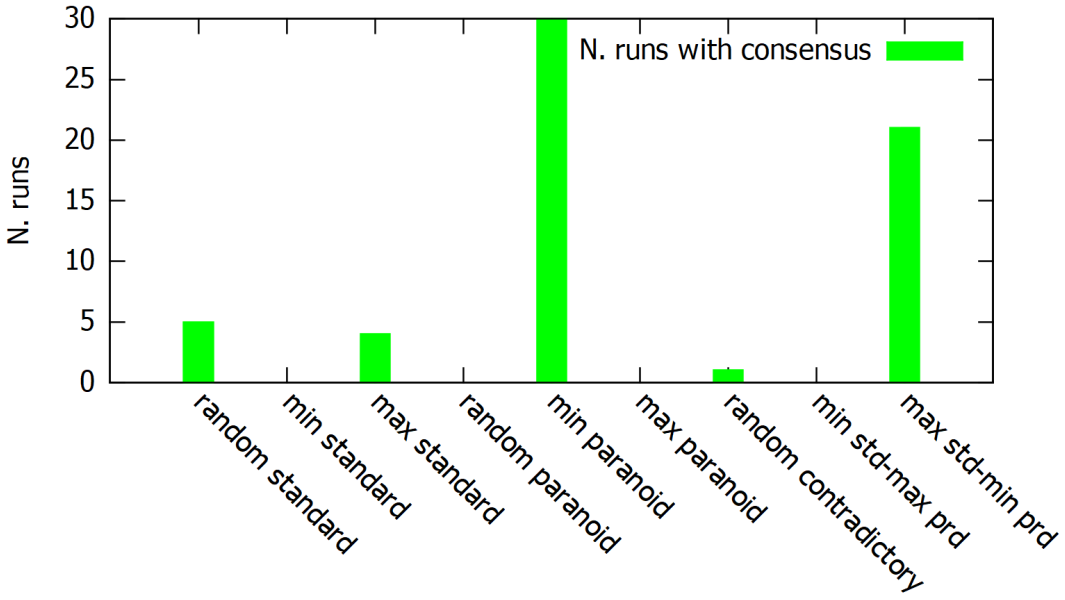


Figure 5.1: Runs with consensus

2. When the nodes seeded are a standard with minimum ranking and a paranoid with maximum ranking, consensus is never reached. However, in many of such runs, the only node holding $\neg p$ is the paranoid one where the initial information is seeded, similarly to the case, described above, when the information is seeded solely in the paranoid node with the maximum ranking.

3. When the nodes seeded are a standard with the maximum ranking and a paranoid with the minimum ranking, consensus on the formula $\neg p$ is reached in 22 runs out of 30. In such situation, the influence of the standard node transmitting p is overwhelmed by the paranoid node.

5.2 Diffusion of conspiracy theories

In this section, we study the diffusion of the "conspiracy theory" $\neg p$. As for consensus we analyze total and scale-free networks. The results concerning the diffusion of $\neg p$ in total networks can be summarized as follows:

1. the diffusion of $\neg p$ concerns all the nodes in the network when the seeded node is paranoid (see section 5.1);

2. There is no such diffusion when the seeded node is standard, since consensus is always reached (see section 5.1);

3. In the case of a contradictory seeding, there is no other parameter which influences the diffusion of $\neg p$ than the order in which the routines are initiated by the code. This means that the diffusion of p, or its negation, depends on which information starts to transmit earlier. Additionally, since the structure of the network in the different configurations is always the same, the results on the diffusion of $\neg p$ are always identical.

Now, we study the diffusion of conspiracy theories in scale-free networks. We dissect the analysis by the three possible seeding of the initial information in the network. Therefore, we analyze the diffusion of $\neg p$ in the following cases:

- when the initial information is seeded in a standard node, i.e. a standard discovery;
- when the initial information is seeded in a paranoid, namely a paranoid discovery;
- when the initial information is contradictory, hence a contradictory discovery.

Standard discovery

In figure 5.2, it is presented the mean of the diffusion of the skewed information $\neg p$ in the possible different network's size and according to the different proportions of paranoid nodes. The diffusion of the skewed information is analyzed by computing the mean value of the number of nodes holding $\neg p$ at the end of a simulation.

Number of nodes	% of paranoids	Mean $\neg p$
50	3%	2.0682
	12%	9.6000
	50%	24.8864
150	3%	8.8636
	12%	33.5909
	50%	74.5341
300	3%	31.6023
	12%	75.9318
	50%	151.3750

Figure 5.2: Average of nodes holding $\neg p$

By looking at figure 5.2, it is clear that the number of nodes that end up holding the deviated information $\neg p$ increases if, while keeping the number of nodes fixed, we raise the proportion of paranoid nodes. Therefore, a large number of paranoid nodes increases the chances of a wider diffusion of the conspiracy theory. In particular, it increases at a much faster rate when the size of the network considered is set at 300 nodes (see figure 5.3). More interestingly, the mean of the diffusion of the skewed information $\neg p$ increases if we keep the proportion of paranoid fixed, while increasing the total number of nodes in a network (see figure 5.4).

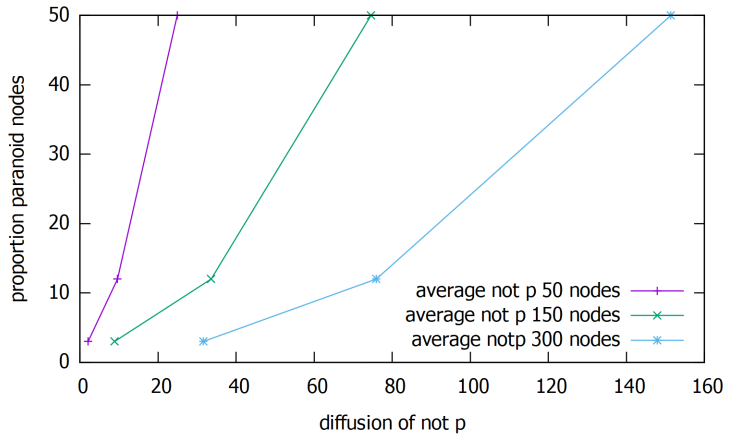


Figure 5.3: proportion of paranoid nodes compared to the diffusion of $\neg p$

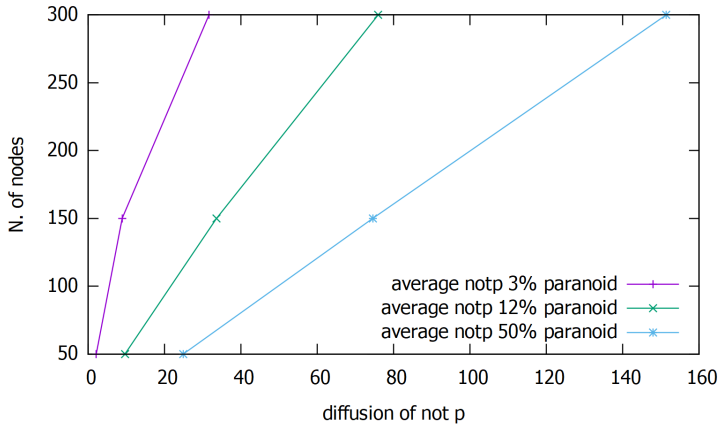


Figure 5.4: Number of nodes compered to the diffusion of $\neg p$

This is in accordance with Madsen et al. (2017), where the simulation conducted, concerning the possibility of growing a Bayesian conspiracy theorist, showed that the diffusion of conspiracy theories is facilitated in larger networks. Indeed, by increasing the number of nodes in a network, the chances of growing echo chamber are higher. In our work, echo chamber are formed, see figure 5.5, when a node holding $\neg p$ controls the flow of the information of a consistent part of the network. In the echo chambers many standard agents, without access to other sources of information, end up believing in the skewed information $\neg p$. This last consideration supports the claim by Madsen et al. (2017); Sunstein & Vermeule (2009), concerning the possibility of endorsing a conspiracy theory when access to appropriate information is prevented.

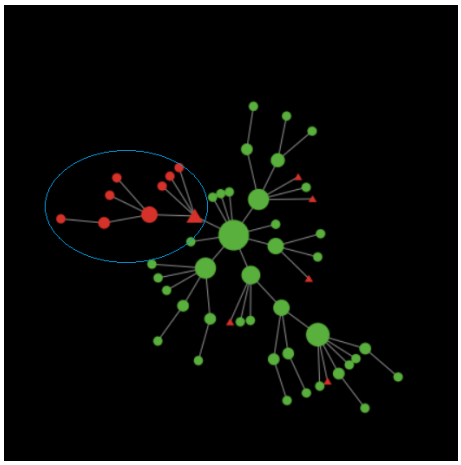


Figure 5.5: An example of echo chamber

Discovery	Number of nodes	% of paranoids	Average $\neg p$
std max	50	3%	2.5333
		12%	9.7000
		50%	24.6667
	150	3%	9.2333
		12%	36.3667
		50%	74.1667
	300	3%	29.9333
		12%	78.3667
		50%	148.7000
std min	50	3%	2.0000
		12%	8.9333
		50%	24.7000
	150	3%	8.1667
		12%	34.2667
		50%	73.8000
	300	3%	31.4000
		12%	71.3333
		50%	152.5667
std random	50	3%	1.5667
		12%	10.1667
		50%	25.2333
	150	3%	9.0333
		12%	30.8000
		50%	75.6333
	300	3%	32.6000
		12%	77.0667
		50%	152.6667

Figure 5.6: Average of nodes holding $\neg p$ in the different discoveries

In figure 5.6, we analyze the influence of the seeded nodes in relation to their

ranking. We consider the mean of the diffusion of $\neg p$ in the different kinds of standard discoveries. However, there seems to be no strong correlation between the ranking of the seeding node and the diffusion of $\neg p$, but only some slight differences (see figures 5.7, 5.8 and 5.9). In general, small differences can be seen when the proportion of paranoids is set at 3% or 12%. Once the proportion is set at 50%, the ranking of the seeding node practically looses all the influence it has on the diffusion of $\neg p$. Hence, the proportion of paranoid nodes itself becomes the crucial parameter.

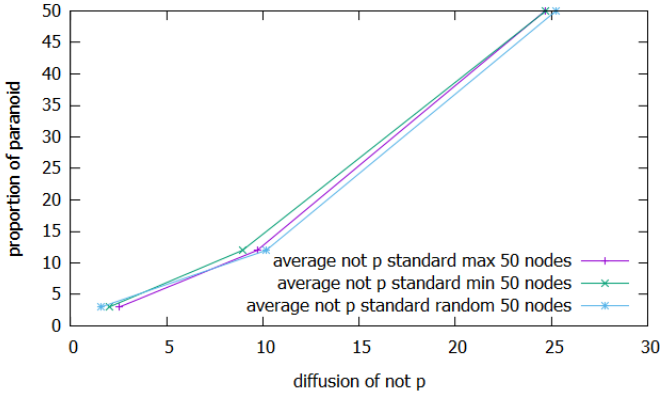


Figure 5.7: Diffusion of $\neg p$ in different discoveries for 50 nodes

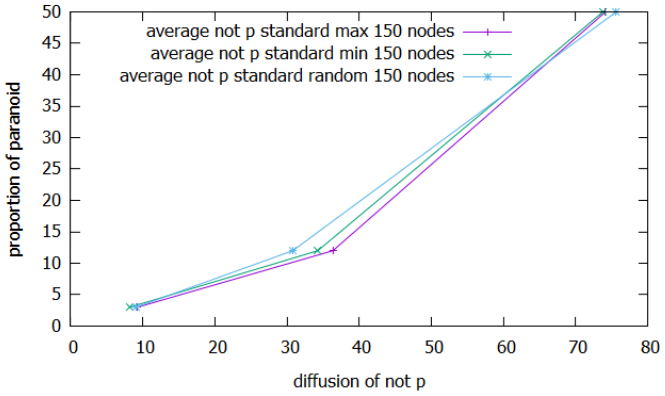


Figure 5.8: Diffusion of $\neg p$ in different discoveries for 150 nodes

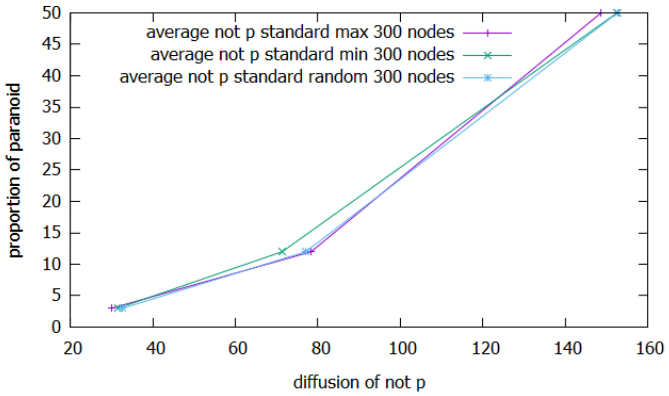


Figure 5.9: Diffusion of $\neg p$ in different discoveries for 300 nodes

Paranoid discovery

In figure 5.10, we analyse the seeding of the initial information of a paranoid node and present the mean value of the diffusion of the skewed information $\neg p$, in the possible different network's size and according to the different proportion of paranoid nodes.

Number of nodes	% of paranoids	Average $\neg p$
50	3%	48.9885
	12%	40.1264
	50%	24.6782
150	3%	138.0227
	12%	114.7727
	50%	75.8182
300	3%	278.4205
	12%	216.7241
	50%	150.6782

Figure 5.10: Average of nodes holding $\neg p$ in paranoid discovey

A difference from the data provided for the discovery standard is that, when the initial information is seeded in a paranoid node, the diffusion of $\neg p$ decreases as the proportion of paranoid nodes increases. In particular, it decreases at a faster rate when the size of the network considered is set at 300 nodes (see figure 5.11). Therefore, in the simulations where the initial information is seeded in a paranoid node, a high number of such nodes in fact reduces the diffusion of $\neg p$. This could be a possible benefit of the paranoid behavior, that should be limited to the cases where distrust towards the authority is actually justified, e.g. for individuals living in dictatorial regimes with no free press (Sunstein & Vermeule, 2009). Indeed, paranoid nodes, differently from standard nodes, have the possibility to "break" echo chamber without requiring access to other information from the one circulating in it. The cause of that lies in the mistrust routine of the paranoid nodes that, in order not to conform with the authority, mistrust its own information when it is consistent with the transmitted one. Instead, the mistrust operation of the standard node has

the benefit of conforming with the current transmitted information, which has its downside in the possibility of growing echo chambers. Hence, it is true that a high number of paranoid nodes could decrease the diffusion of $\neg p$, but at the price of radicalising the polarization of the opinion in the network.

For the standard discovery, the mean of the diffusion of the skewed information $\neg p$ increases if we keep the proportion of paranoid nodes fixed, while increasing the total number of nodes in a network (see figure 5.12). However, in accordance with what is explained above, and opposed to the standard discovery, the diffusion of $\neg p$, in relation to the number of nodes, increases at a much faster rate when the proportion of paranoid is low.

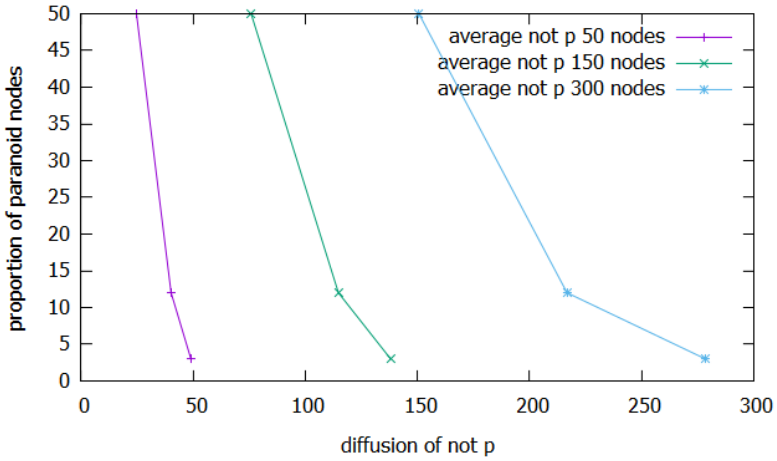


Figure 5.11: Diffusion of $\neg p$ compared to the proportion of paranoid nodes in paranoid discovery

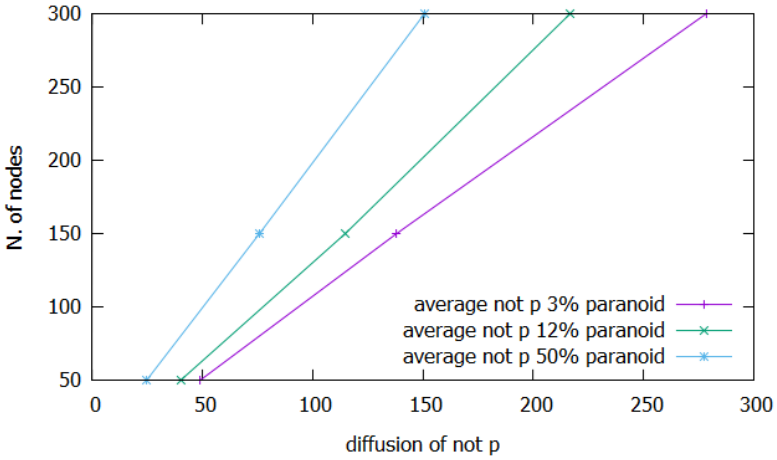


Figure 5.12: Diffusion of $\neg p$ compared to the number of paranoid nodes in paranoid discovery

We analyzed the influence of the ranking of the node seeded with the initial information, considering the mean value of the diffusion of $\neg p$ in the different kind of paranoid discoveries. However, as for standard discovery is concerned, no strong

correlation can be found between the ranking of the seeding node and the diffusion of $\neg p$, but only some slight differences (see figures 5.13, 5.14 and 5.15).

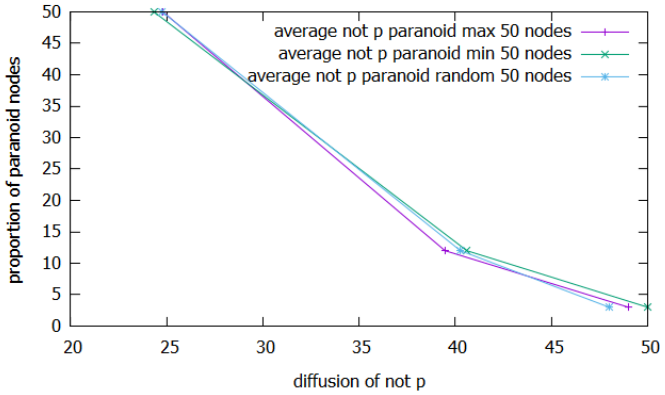


Figure 5.13: Average diffusion of $\neg p$ for 50 nodes in different paranoid discoveries

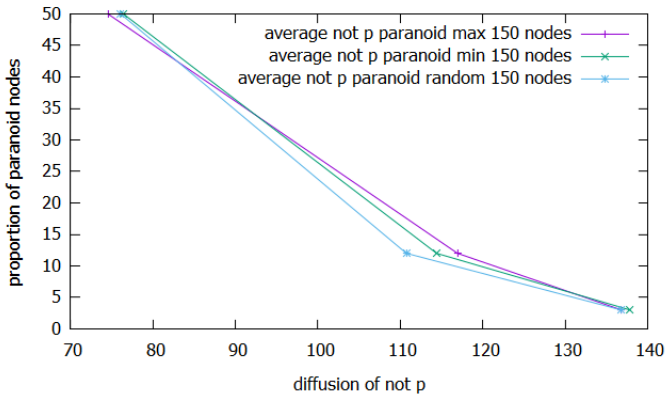


Figure 5.14: Average diffusion of $\neg p$ for 150 nodes in different paranoid discoveries

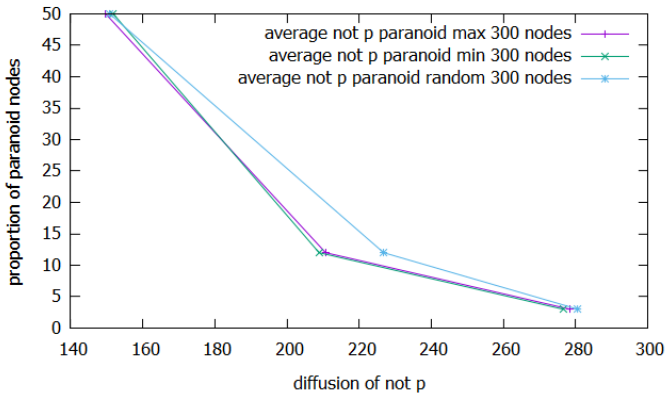


Figure 5.15: Average diffusion of $\neg p$ for 300 nodes in different paranoid discoveries

Contradictory discovery

In this section we present the results concerning the seeding in the network of contradictory information. We calculate the mean value of the diffusion of $\neg p$ in the three different considered contradictory discoveries:

1. the seeding of p in the standard node with the maximum ranking and the seeding of $\neg p$ in the paranoid node with the minimum ranking;
2. the seeding of p in the standard node with the minimum ranking and the seeding of $\neg p$ in the paranoid node with the maximum ranking;
3. the seeding of p in a random standard node and the seeding of $\neg p$ in a random paranoid node.

Discovery	Number of nodes	% of paranoids	Average $\neg p$
std max-prd min	50	3%	46.1000
		12%	37.0667
		50%	24.9000
	150	3%	124.4333
		12%	115.4333
		50%	75.4667
	300	3%	272.3667
		12%	216.1000
		50%	149.1667
std min-prd max	50	3%	4.6667
		12%	15.1333
		50%	23.7333
	150	3%	18.2667
		12%	42.5333
		50%	74.4333
	300	3%	39.1333
		12%	73.5000
		50%	146.5667
random	50	3%	19.2667
		12%	22.9000
		50%	24.4333
	150	3%	71.7333
		12%	88.1667
		50%	75.0000
	300	3%	105.4000
		12%	130.6333
		50%	149.8333

Figure 5.16: Average of nodes holding $\neg p$ in contradictory discoveries

For all the contradictory discoveries, we found out that the value becomes nearly identical when the proportion of paranoid nodes is set at 50%. As explained before,

this is due to the fact that the ranking of the seeding node practically looses all the influence it has on the diffusion of $\neg p$ once the proportion of paranoid nodes increases, and the latter becomes the crucial parameter. For what concerns the proportion of paranoid nodes set at 3% and 12%, the different seeding acts differently (see figure 5.17, 5.18 and 5.19):

1. The discovery in which the initial information is seeded in the standard node with the maximum ranking and in the paranoid node with the minimum ranking shows the lowest value concerning the diffusion of $\neg p$ in all the different network's sizes. In addition, the diffusion of the conspiracy theory increases as the proportion of paranoid nodes rises;

2. The discovery in which the initial information is seeded in the standard node with the minimum ranking and in the paranoid node with the maximum ranking performs the worst if the goal is to limit the diffusion of $\neg p$. However, the diffusion of the conspiracy theory decreases as the proportion of paranoid nodes increases.

3. The random contradictory discovery performs worse than the discovery where the seeded standard node is the minimum and the seeded paranoid node is the maximum. Although, it performs better than the discovery where the seeded standard node is the maximum and the seeded paranoid node is the minimum. The mean value of the diffusion of $\neg p$ increases or decreases depending on the network's size. In particular, it slightly increases when the size is set at 50 and 300 nodes, while slightly decreases when the size is set at 150 nodes.

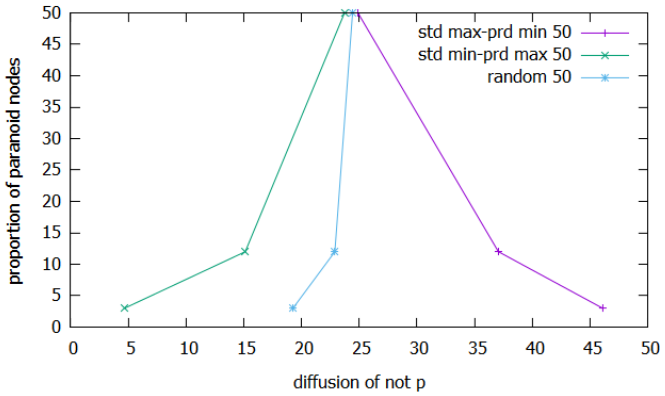


Figure 5.17: Average diffusion of $\neg p$ for 50 nodes in contradictory discoveries

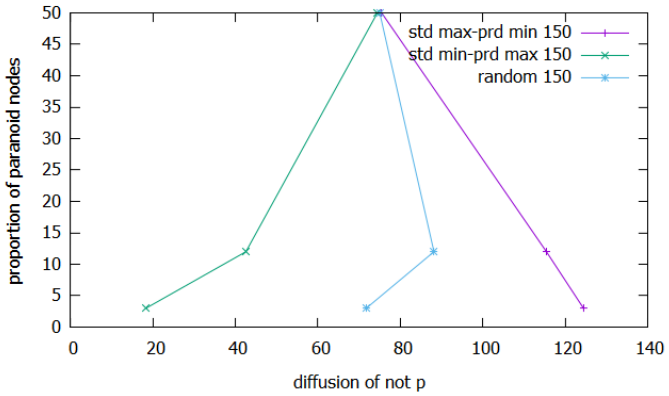


Figure 5.18: Average diffusion of $\neg p$ for 150 nodes in contradictory discoveries

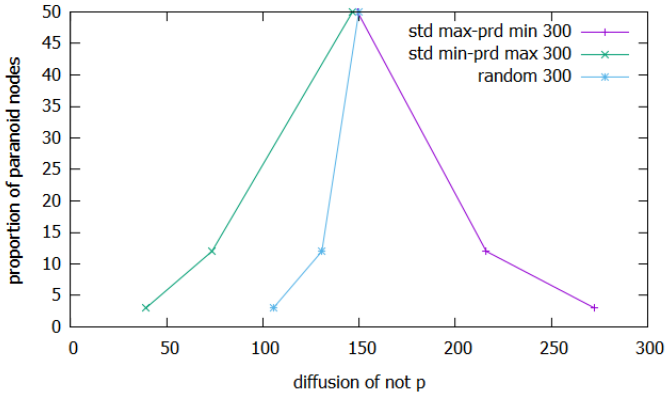


Figure 5.19: Average diffusion of $\neg p$ for 300 nodes in contradictory discoveries

Betweenness centrality

Since we did not find any strong correlation between the ranking of the seeding node and the diffusion of $\neg p$, a crucial factor that predicts the diffusion of $\neg p$ is the presence of echo chambers, where the only accessible information to the nodes is the conspiracy theory. Therefore, such factor concerns more the network structure than the ranking of the seeding nodes, which depends on the epistemic attitudes of the nodes. Hence, to take into account the presence of echo chambers of nodes holding $\neg p$, we calculated the betweenness centrality of nodes holding $\neg p$ that are also linked with others nodes holding $\neg p$. We investigate its validity by looking at anomalous runs of the simulation where the diffusion of $\neg p$ is high. To spot such runs, we calculate the standard deviation of $\neg p$ in the different possible configurations concerning the discovery in which the seeded node is a random standard one. Then we looked for the runs where the diffusion of $\neg p$ is above the upper bound of the standard deviation. At this point, we compared the betweenness centrality of such anomalous run with the mean value of the betweenness centrality of the considered particular configuration. The results of this analysis are introduced in figures 5.20, 5.21 and 5.22

Betweenness centrality 50 nodes			
% of paranoid nodes	Nodes with not p skewed runs	sum Betweenness	Mean betweenness
3%	5 13	233 772	44.6333
12%	16 17 17 20 26	1079 1106 790 2136 2349	447.6333
50%	29 29 29 29 29 30	2438 2847 1713 570 1039 2204	1252.8000

Figure 5.20: Betweenness centrality 50 nodes

Betweenness centrality 150 nodes			
% of paranoid nodes	Nodes with not p skewed runs	sum Betweenness	Mean betweenness
3%	18 31 46	4665 9008 151888	1443.4667
12%	44 45 60 76	14415 9457 15239 26133	4773.4000
50%	82 82 82 83 84 85 88	4981 13189 12606 26929 14231 10915 17523	16055.2333

Figure 5.21: Betweenness centrality 150 nodes

Betweenness centrality 300 nodes			
% of paranoid nodes	Nodes with not p skewed runs	sum Betweenness	Mean betweenness
3%	81 87 228	43062 54325 187437	15622.2000
12%	121 121 137 210	74708 71179 115557 202918	35248.5000
50%	162 162 165 165 163 170	66323 106839 107205 75950 103909 63692	90674.6000

Figure 5.22: Betweenness centrality 300 nodes

The main findings are:

- For what concerns the proportion of paranoid nodes of 3% and 12%, the anomalous runs, where the diffusion of $\neg p$ is high, show a correlation with high value of the betweenness centrality, which are way above the average;
- For what concerns the proportion of paranoid nodes set at 50%, it is not always the case that to an anomalous run of the simulation corresponds anomalous high betweenness centrality. The reason is the one explained also before: a higher number of paranoids nodes actually decreases the chances of growing echo chambers, since their attitude is more prone to favour the polarization of information;
- The sum of the betweenness centrality is also a parameter that could justify the skewed data in the runs of the simulation: an high value of $\neg p$ depends in particular networks on the possibility of growing echo chambers.

5.3 Conclusions

We summarise hereafter the main results from our analysis concerning the diffusion of conspiracy theories. For what concerns consensus, the paranoid behaviour is a hurdle for consensus reaching transmission; consensus is reached solely in some runs of the simulation concerning a network with a dimension of 50 nodes and with the proportion of paranoid nodes set at 3%. Interestingly, in just one setting consensus is reached in every repetition of the simulation: when the initial information is seeded in the paranoid node placed at the top of the hierarchy. In this case, the only paranoid node is the leading authority, therefore no operation of mistrust will impede consensus.

As expected, an increasing number of paranoid nodes in a network increases the diffusion of the "false" information. In particular, it increases at a much faster rate when the size of the network considered is set at 300 nodes.

We noticed that larger networks facilitate the diffusion of the skewed belief. In accordance with results presented in Madsen et al. (2017), large networks increase the possibility of growing echo chambers where the only information available to the standard agents is the skewed one. Therefore, without having other sources of information, many standard agents end up believing in a conspiracy theory. This last considerations support the claim sustained by Madsen et al. (2017); Sunstein & Vermeule (2009), concerning the possibility of endorsing a conspiracy theory when access to appropriate information is prevented.

We found no strong correlation between the diffusion of $\neg p$ and the position in the hierarchy of the seeding node. This was expected to be a parameter that would predict the diffusion of the false information, in particular concerning anomalous runs of the simulation where the number of agents endorsing $\neg p$ was particularly high. In these last cases we found that a more clarifying parameter was the betweenness centrality of nodes holding $\neg p$, which increases the probability of growing echo chambers where the only information available is the deviated one. However the ranking of the seeded nodes in contradictory discoveries, along with their epistemic characterization, influences the diffusion of $\neg p$ when the proportion of paranoids node is not high.

In the simulations where the initial information is seeded in a paranoid node, a high number of such nodes in fact reduces the diffusion of $\neg p$. This could be a possible benefit of the paranoid behavior, that should be limited to the cases where distrust towards the authority is actually justified, e.g. for individuals living in dictatorial regimes with no free press (Sunstein & Vermeule, 2009). Related to this, we found out that a very high proportion of paranoid nodes decreases the chances of growing echo chambers. However, a high number of such nodes maximise the polarization of the information.

From these results we could make some reflections concerning possible ways to limit the transmission of conspiracy theories. For the agents we defined as standard, their endorsing of conspiracy theories lies in the lacking of the appropriate information. As it has been showed, this situation could be common, since large networks increase the probability of growing echo chambers where the only available information is the deviated one. This also happens in real case scenario such as in social media, where phenomena similar to the echo chambers we described may arise (Zannettou et al., 2018). An intuitive solution to this problem could lie in the exposure to more diverse information which may seem a solution at hands considering the means of accessing information we possess. However, it may not be easy. In Bakshy et al. (2015) the exposure to ideological cross cutting content by Facebook users is examined and it is shown to be more correlated with individual choices than with the platform's algorithms, suspected by many as being the cause of users' homophily. Ultimately, the solution to the problem may lie in a more incisive action regarding the education of the population toward a correct use of the information. However, this may be something not easily achievable.

More thorny issues arise when we consider the agents we defined as paranoid. Their attitude may have some possible benefit regarding situations where the au-

thority is actually unreliable. However, in more common situation this is not the case, and their widespread distrust is inappropriate. Their problem lies in the low trust levels towards what they perceive as authorities, a problem which cannot be easily addressed. One strategy that our simple model suggests may rely on the way in which sources of information present themselves: a less distant approach may have some benefits.

5.4 Further work

The present work could be extended and improved. An immediate refinement of the analysis could be achieved by including other networks' topology such as linear and random. In this way we could also study in these setting how the paranoid agents spread conspiracy theories and affect consensus. Another improvement, given the key role of echo chambers in the diffusion of the skewed information related to the standard behaviour, could be analysing the clustering of standard nodes. Additionally, in the experimental study we considered only the mean value of the nodes holding $\neg p$, since we thought that skewed value were variables of interest. It would be interesting to see how these values change if we consider also their median. Finally, it would be interesting to introduce in the logic derivations that take into account not only the behaviour of the agent receiving the information but also the behaviour of the agents sending it. For example, by introducing rules that initiate a particular behaviour when a paranoid agent reads information from another paranoid agent, that would act differently if the same message was sent by a standard one.

Bibliography

Abalakina-Paap M., Stephan W. G., Craig T., Gregory W. L., 1999, *Beliefs in Conspiracies*, Political Psychology, 20, 637

Bakshy E., Messing S., Adamic L. A., 2015, *Exposure to ideologically diverse news and opinion on Facebook*, Science, 348, 1130

Bruder M., Haffke P., Neave N., Nouripanah N., Imhoff R., 2013, *Measuring Individual Differences in Generic Beliefs in Conspiracy Theories Across Cultures: Conspiracy Mentality Questionnaire*, Frontiers in Psychology, 4, 225

Cohnitz D., 2017, *Critical Citizens or Paranoid Nutcases: On the Epistemology of Conspiracy Theories*. Utrecht: Universiteit Utrecht, Faculteit Geesteswetenschappen

Douglas K., Uscinski J., Sutton R. M., Cichocka A., Nefes T., Ang C. S., Deravi F., 2019, *Understanding conspiracy theories*, Advances in Political Psychology, 40, 3

Freeman L. C., 1977, *A Set of Measures of Centrality Based on Betweenness*, Sociometry, 40, 35

Goertzel T., 1994, *Belief in Conspiracy Theories*, Political Psychology, 15, 731

Golbeck J., 2013, in Golbeck J., ed., , Analyzing the Social Web. Morgan Kaufmann, Boston, pp 25 – 44, doi:https://doi.org/10.1016/B978-0-12-405531-5.00003-1, http://www.sciencedirect.com/science/article/pii/B9780124055315000031

Law 2003, in Proceedings of the 2003 Winter Simulation Conference, 2003.. pp 66–70 Vol.1, doi:10.1109/WSC.2003.1261409

Macklin G., 2019, *The Christchurch Attacks: Livestream Terror in the Viral Video Age*, CTC Sentinel, 12

Madsen J., Bailey R., Pilditch T., 2017.

Martínez R., Casacuberta D., Figueras C., 1999, *The R files: applying relevance model to conspiracy theory fallacies*, Journal of English Studies, ISSN 1576-6357, Nº 1, 1999, pags. 45-56, 1

Mashuri A., Zaduqisti E., 2014, *We believe in your conspiracy if we distrust you: the role of intergroup distrust in structuring the effect of Islamic identification, competitive victimhood, and group incompatibility on belief in a conspiracy theory*, Journal of Tropical Psychology, 4, e11

Miller J. M., Saunders K. L., Farhart C. E., 2016, *Conspiracy Endorsement as Motivated Reasoning: The Moderating Roles of Political Knowledge and Trust*, American Journal of Political Science, 60, 824

Oliver J. E., Wood T. J., 2014, *Conspiracy Theories and the Paranoid Style(s) of Mass Opinion*, American Journal of Political Science, 58, 952

Parsons S., Simmons W., Shinhoster F., Kilburn J., 1999, *A test of the grapevine: An empirical examination of conspiracy theories among African Americans*, Sociological Spectrum, 19, 201

Primiero G., 2019, *A Logic of Negative Trust*

Primiero G., Raimondi F., Bottone M., Tagliabue J., 2017, *Trust and distrust in contradictory information transmission*, Applied Network Science, 2, 12

Sunstein C. R., Vermeule A., 2009, *Conspiracy Theories: Causes and Cures**, Journal of Political Philosophy, 17, 202

Zannettou S., Bradlyn B., De Cristofaro E., Kwak H., Sirivianos M., Stringini G., Blackburn J., 2018, in Companion Proceedings of the The Web Conference 2018. WWW '18. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, p. 1007–1014, doi:10.1145/3184558.3191531, https://doi.org/10.1145/3184558.3191531