Contradictory information flow in networks with trust and distrust

Giuseppe Primiero and Michele Bottone and Franco Raimondi and Jacopo Tagliabue

Abstract We offer a proof system and a NetLogo simulation for trust and distrust in networks where contradictory information is shared by ranked lazy and sceptic agents. Trust and its negative are defined as properties of edges: the former is required when a message is passed bottom-up in the hierarchy or received by a sceptic agent; the latter is attributed to channels that require contradiction resolution, or whose terminal is a lazy agent. These procedures are associated with epistemic costs, respectively for confirmation and refutation. We describe the logic, illustrate the algorithms implemented in the model and then focus on experimental results concerning the analysis of epistemic costs, the role of the agents' epistemic attitude on distrust distribution and the influence of (dis)trust in reaching consensus.

1 Introduction

Trust of information transmissions facilitates reliability and enforces security in networks. This applies in particular to hierarchical structures, e.g. in access control models [3, 2, 14], and where reputation is at work, e.g. in social networks [9, 18, 4]. Trust and distrust on communication channels are also affected by the agents' epis-

Giuseppe Primiero
Department of Computer Science, Middlesex University London, e-mail: G.Primiero@mdx.ac.uk

Michele Bottone
Department of Computer Science, Middlesex University London, e-mail: M.Bottone@mdx.ac.uk

Franco Raimondi
Department of Computer Science, Middlesex University London, e-mail: F.Raimondi@mdx.ac.uk

Jacopo Tagliabue
AXON VIBE, New York, e-mail: tagliabue.jacopo@gmail.com

Preprint of a Paper to appear in H. Cherifi, S. Gaito, W. Quattrociocchi, A. Sala (eds.), *Complex Networks & Their Applications V*, Studies in Computational Intelligence, Springer, forthcoming.

temic attitude, their ability and willingness to check information and their readiness to reject it. Negative trust has recently become a topic of interest in computational contexts [13, 11]. In particular, understanding conditions of (dis)trust propagation and the costs related to topological and epistemic factors is crucial for dynamic (social) network analysis and access control models [1, 6, 21, 8], with applications in mathematics, computer science, economics and biology. Negative accounts of trust are essential especially for networks that allow contradictory information diffusion but require coherent agents.

In this paper we offer a logic and a NetLogo simulation for networks with contradictory information and where agents identify their channels as trustful or distrustful. Our agents are qualified as sceptic or lazy and are given an initial ranking depending on the topological features of the network. The network is seeded initially with two items of contradictory information $(p, \neg p)$. Each node is labelled by either piece of data, with a resolution procedure when both are received by the same node. At each step, the node assigns a trust or a distrust property to the relevant edge. In our experimental analysis we consider in particular:

1. the epistemic costs of trust and distrust according to different network topologies;
2. the distrust distribution in view of the epistemic attitude of the seeding agents;
3. the role of distrust in reaching consensus.

The paper is organized as follows. In Section 2 we overview related work. In Section 3 we introduce the calculus $\texttt{(Un)SecureND}^{sim}$ which includes rules for trust and distrust. In Section 4 we provide the principles underlying the graph construction and algorithm design at the basis of the simulation. In Section 5 we describe our experimental results. Finally, Section 6 presents general observations on our analysis and shortly illustrates future work.

2 Related Work

In reporting on previous work, we focus in particular on three different aspects: controversial users vs. controversial trust values; binary and continuous trust values; local vs. global trust methods.

In [12] controversial users are those generating a disagreement on their trustworthiness, either as the minimum between trust and distrust evaluations by other users, or as the difference in the number of trust and distrust judgements. [20] considers instead controversial trust values between two nodes, determined either as the trust weight of their edge, or as a fixed negative value when no path exists, or as a continuous value $t \in [0, 1]$ when there is no direct edge. Similarly, in our logic trust is a function on formulas obtained by verification, mimicked in the network model by a property of edges when a node is labelled.

Differently from the above, our model uses discrete values but it combines the comparative ranking of agents with both their epistemic attitudes and a majority

selection in the case of conflicting information. [12] also uses a binary classification for users, so do several models for belief diffusion in social networks, with binary opinions for agents, considering neighbours' influence [5, 9] or majority ([18]). Continuous models, on the other hand, might depend on the weight of other agents' opinion [10] or admit influence only below a certain distance [7].

Trust defined by global methods is a value attached to a user and appropriate for a reputation evaluation at network level; in local methods, trust is inferred instead as a value between source and sink nodes, i.e. it is an edge feature. As it appears clearly from the above, our approach uses a local trust method in the case of non-conflicting information, resorting to a computation of trust path lengths to determine which elements need to be distrusted in the case of conflicting information. This combination of features recalls the two controversial cases discussed in [20]: the *ToTrustOrNotToTrust* case resembles our binary choice, but moderated by continuous trust values, while we rely on ranking and epistemic attitudes; the *Asymmetric Controversy* case resorts to path lengths with preference for shortest paths, while we base our result on the number of distrustful edges present in each path.

To the best of our knowledge, no other work in the current literature combines a rule-based semantics with ranked agents with epistemic attitude, using local trust values with path length analysis for the resolution of contradictory information.

3 (Un)SecureND^{sim}

The natural deduction calculus SecureND [16] is a logic designed for secure operations on resources issued by subjects with different privileges; it guarantees trusted content checked for consistency at every transmission. (Un)SecureND [15] is an extension with negation to model two forms of negative trust. In [17], the calculus SecureND^{sim} is adapted to model contradictory information propagation under trust in a network of ranked agents and is simulated in NetLogo [19]. In this contribution we present (Un)SecureND^{sim}, extending the previous system to deal with a distrust function. We refer to a set of agents as V and an individual agent as v_i. Agents behave differently in the context of information transmission:

- *sceptic agents* and *agents reading from below in the hierarchy* require verification when receiving a message, and as a result they trust the related channel;
- *lazy agents* and *all agents in the presence of contradictions* have a rejection attitude, with the result of distrusting the related channel.

Verification and rejection are computationally costly processes for the agents and these costs are tracked in our model.

Definition 1. The syntax of (Un)SecureND^{sim} is defined by the following alphabet:

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

V is the set containing lazy and sceptic agents; BF^V are literals, i.e. atoms and their negations; in the following, when needing a metavariable for either, we will use ϕ^V; *mode* is for access functions over atoms; RES^V includes both contents and access modes, with negation. In line with standard notation for natural deduction, we use Γ^V to express a context of expressions (typed by one agent in V, and feasible to extension to another agent's context) in which a given formula is derivable: such a context matches the graph G of agents introduced in the next section; the derived formula matches a new labelled vertex added to the graph. Formulas of this language are of the general form $\Gamma^{v_i} \vdash RES(\phi^{v_j})$, saying that an agent v_i accesses under her profile a message ϕ originated by agent v_j. Access is here neutral for all the operations included in *mode*. An order relation $\leq$ over $V \times V$ models the dominance relation between agents: $v_i \leq v_j$ means that agent v_i has equal or higher priority (e.g. in terms of security privileges) than agent v_j.

The rules system $\texttt{(Un)SecureND}^{sim}$ is introduced in Figure 1 and it assumes that $v_i \leq v_j$ holds. This logic allows the following operations. Any content is accessible within a well-formed (wf) user profile (*Atom*). Accessing a negation of a content implies that the contrary cannot be accessed ($\neg$-distribution): this is a strong negation rule, justifying the resolution procedure for contradictions. Any content can be read from agents downwards in the order relation (*read_down*) and it is accepted if it preserves the profile consistency (*read_elim*). Reading by an agent upwards in the dominance relation or by a sceptic agent is possible by invoking a verification procedure (*verify_high* and *verify_sceptic* respectively). Such verification checks consistency and then applies a trust function on the object of the message (*trust*). Reading and trusting guarantee rights to write formulae (*write_trust*). The remaining rules define the behaviour of distrust relations. Reading contradictory information or reading by a lazy agent induce a rejection procedure (*unverified_contra* and *unverified_lazy* respectively). This in turn means that a distrust operation is executed (*distrust*), and the opposite message to the one read can be written (*distrust_elim*).

4 Model Design and Implementation

The network is an undirected graph $G = (V, E)$, with a set $V = \{v_i, \ldots, v_n\}$ of vertices (agents) and a set $E = \{e_{(i,j)}, \ldots, e_{(n,m)}\}$ of edges (information transmission channels). A labelled node $v(p)$ denotes an agent knowing p; $v(\neg p)$ denotes an agent knowing $\neg p$; $v()$ is used for a vertex with no label and denotes an agent who does not hold any knowledge yet. An edge between two nodes is fully denoted by $e(v_i(), v_j())$ with the appropriate labels: $e(v_i(p), v_j())$ expresses a channel from i

$$\frac{\Gamma^{v_i} \vdash wf}{\Gamma^{v_i};\Gamma^{v_j} \vdash \phi^{v_j}}\ \text{Atom} \qquad \frac{\Gamma^{v_i} \vdash mode(\neg\phi^{v_j})}{\Gamma^{v_i} \vdash \neg mode(\phi^{v_j})}\ \neg\text{-distribution}$$

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

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

$$\frac{\Gamma^{v_i} \vdash Read(\phi^{v_j})}{\Gamma^{v_i} \vdash Verify(\phi^{v_j})}\ verify\_high \qquad \frac{\Gamma^{v_i} \vdash \phi^{v_i} \qquad v_j \in \texttt{sceptic\_node}}{\Gamma^{v_j} \vdash Verify(\phi^{v_i})}\ verify\_sceptic$$

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

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

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

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

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

Fig. 1 The system (Un)$\texttt{SecureND}^{sim}$

to j such that the former can transmit p over. A non-standard notation with three nodes $e(v_i(p), v_j(), v_k(\neg p))$ is used to abbreviate the fact that the following edges exist: $e(v_i(p), v_j())$ and $e(v_j(), v_k(\neg p))$ and it requires a resolution procedure. When need for reference to multiple vertexes arises, we shall use the notation $v_{i,\ldots n}$. The order relation among nodes is total or partial in view of the network topology. In a total network, each vertex has an edge connecting it to any other vertex and all have equal ranking; the underlying dominance relation is then a total order. In the linear network, each vertex has an edge to the next vertex higher in the ranking; by transitivity, also this order is total. In the random network, by introducing a new node at least one edge with another vertex is established; the ranking is here assigned by the seeding node and never overwritten, the order is partial. The scale-free network model uses the Barabasi-Albert method: it is initialised by $m = 3$ nodes and each node v_j without neighbours is connected to up to $n < m$ existing vertices with a probability $p_{v_j} = \frac{k_{v_j}}{\sum_{v_i} k_{v_i}}$, 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

```
PROCEDURE Transmission(G), with φ ∈ BF^V

G := (V,E)

FOR e(v_i(φ), v_j()) ∈ G
    IF v_j() ∈ sceptic 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
        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
```

Fig. 2 Algorithm for Simple Information Transmission

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

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

Fig. 3 Algorithm for Trust Costs Increase

that already have a high number of links. The ranking in this case is given as $\frac{1}{|edges|}$. The maximum number of vertices in our graphs is set at 300.

The randomly seeded contradictory information $(p, \neg p)$ flows in the network, according to the algorithm `Transmission` in Figure 2. If the receiving agent is `sceptic` or a non-contradictory message comes from below in the dominance relation, a successful transmission is preceded by a sub-routine `Verify`, described in Figure 3. The latter implies an epistemic cost, the new node is successfully labelled and the edge is qualified as trusted. If the receiving agent is `lazy`, a new subroutine `Distrust` is executed, by which the edge is qualified as distrusted and the related epistemic costs are increased, Figure 4. A node receiving contradictory data $(p, \neg p)$ starts a resolution process `SolveConflict`, see Figure 5: it analyses the number of distrusted links appended to each neighbour with each contradictory piece of information and it selects the new label from the least distrusted one, proceeding by random choice $(*)$ when an equal number of distrusted links is detected. It then executes the subroutine `Distrust` on the selected link.

```
1  PROCEDURE Distrust(e(v_i(φ), v_j()))
2
3   set COSTDISTRUST+1
4   set DISTRUSTLINK e(v_i(φ), v_j(¬φ))
5  RETURN Trusted(G)
6  ENDPROCEDURE
```

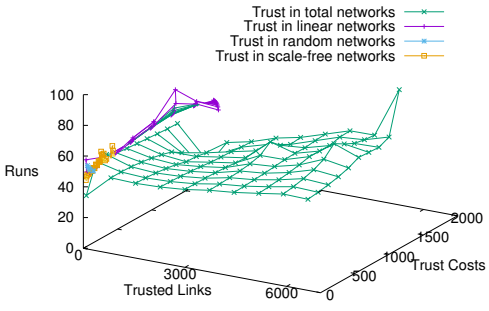
Fig. 4 Algorithm for Distrust Costs Increase

```
1  PROCEDURE SolveConflict(e(v_i(φ), v_j(), v_k(¬φ)))
2
3   let d1 #DISTRUSTLINK e(v_{i,...n}(φ), v_j())
4   let d2 #DISTRUSTLINK e(v_{k,...m}(¬φ), v_j())
5
6  IF (length d1 > length d2)
7     THEN G' := G ∪ (v_j(¬φ)) AND Distrust(e(v_i(φ), v_j(¬φ)))
8  ENDIF
9
10 IF (length d1 < length d2)
11      THEN G' := G ∪ (v_j(φ)) AND Distrust(e(v_k(¬φ), v_j(φ)))
12 ENDIF
13
14 IF (length d1 = length d2)
15    IF *
16    THEN G' := G ∪ (v_j(¬φ)) AND Distrust(e(v_i(φ), v_j(¬φ)))
17    ELSE G' := G ∪ (v_j(φ)) AND Distrust(e(v_k(¬φ), v_j(φ)))
18    ENDIFELSE
19 ENDIF
20 ENDPROCEDURE
```

Fig. 5 Algorithm for Conflict Resolution

5 Experimental results

The code for the simulation and all data from the experiments are available at https://bitbucket.org/gprimiero/cn16. The experiments have been executed on a machine with 7.7 GB of memory, 64bit Ubuntu 16.04 system, NetLogo 5.3. We have collected data from several synthetic networks of fixed dimensions between 10 and 300 nodes, with seeding of labels $(p, \neg p)$ randomly associated to two sceptic/lazy nodes. We consider first different network topologies and then focus on scale-free networks only, which better represent the topology of complex graphs as they occur for example in social networks. On the other hand, linear networks are more common in hierarchical structures that can be encountered in conditions of access control. In both cases, the role of trust and distrust operation is crucial to information propagation.



	Average Trusted Links	Average Trust Cost
Random	47.78	940.92
Linear	77.78	434.16
Scale-free	102.21	415.42
Total	971.44	385.11

Fig. 6 Trust distribution and average costs

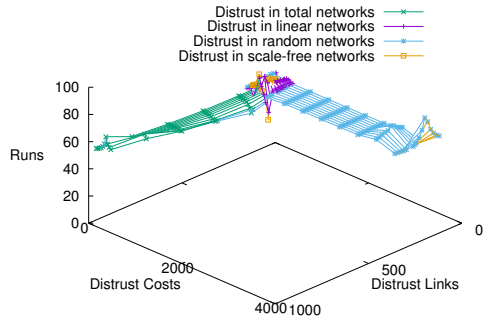
5.1 Costs of Trust/Distrust by Network Topology

In the first run of experiments we compare different network topologies of fixed size (300 nodes), each equipped with a fixed proportion of sceptic nodes (50%). We consider in particular the size of trusted and distrusted edges and the related costs for each topology.

As shown in Figure 6 and the associated Table, the average rate of links and costs is inversely proportional: the former increases from random, through linear, scale-free and total networks, while the latter decreases. Given the fixed number of sceptic agents across the various topologies, the decrease in costs should be mainly associated with the ranking of agents and their order, while the increase in trusted links is purely due to the number of links in the network. From these data it appears that random networks perform the worst, as the required costs are high but the obtained links are less than in scale-free or linear networks.

The different topologies show a similar pattern with respect to distrust values. As shown in Figure 7 and the associated Table of average values, random networks are the most expensive with respect to distrust, and have the lowest number of distrusted links; linear networks remain constrained in number of distrusted links, with costs decreasing; scale-free networks do not show a sensibly better behaviour, with comparable number of distrusted links and costs; finally, total networks perform the best, with the highest levels of links and relatively lower costs. As shown in the graph, it is remarkable the diverging behaviours of total and random networks: the former ones have almost stable distrust cost with increasing distrusted links, while the latter have stable links with increasing costs.

The comparison between tables shows that the average number of trusted and distrusted links grows in parallel, while the related costs decrease in a similar vein across the different topologies. Trust propagates a lot more than distrust in these balanced networks, suggesting that the former is a more frequent and more relevant property in information transmission than the latter.

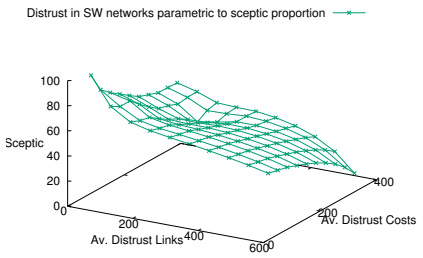


	Average Distrust Links	Average Distrust Cost
Random	35.4	167.38
Linear	71.82	130.18
Scale-free	75.29	134.5
Total	264.13	103.1

Fig. 7 Distrust distribution and average costs

5.2 Distrust and epistemic attitude

In this and the following experiments, we focus on scale-free networks only and their distrust behaviour.[1] First, we consider distrust as a parameter of the proportion of lazy agents in a network of 300 nodes, with a random assignment of seeds to agents. As shown in Figure 8, there is a strict correlation between the proportion of sceptic and the distrust behaviour: the more lazy agents are present in the network, the higher its overall distrust value. While this is obvious in view of the algorithm design, it is interesting to remark that in the case of a fully sceptic network (where no lazy agents are allowed), the value of distrust is to be associated entirely with the presence of contradictory information, and hence it can be used as a parameter of contradiction diffusion. The associated Table offers average values over 100 runs. It illustrates that conflict resolution is responsible on average for roughly 10% of the network's distrusted edges, with costs averaging at around $\frac{1}{7}$ of those of a highly lazy network (i.e. with 10% of sceptic agents).



% Sceptics	Av. Distrusted Links	Av. Distrust Cost
10%	575.45	350.92
30%	170.11	276.79
50%	129.98	213.36
80%	69.2	113.83
100%	32.25	45.35

Fig. 8 Distrust behaviour and epistemic attitude.

[1] For a more detailed analysis of further aspects of trust behaviour, see [17].

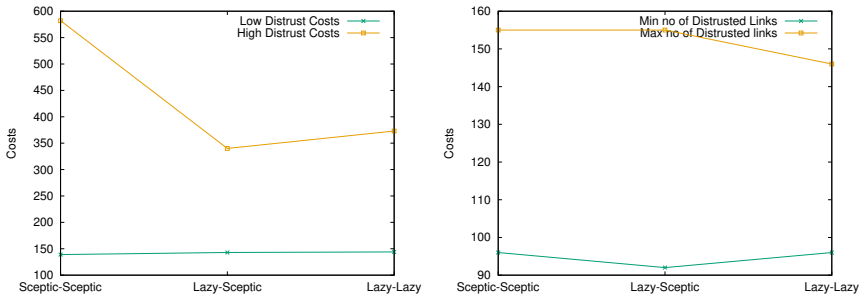


Fig. 9 Initial nodes' epistemic attitudes and distrust

We now extract the values for a balanced network (i.e. with 50% of sceptic agents) and compare them to the initial distribution of seeds qualified as lazy-sceptic agents. As Figure 9 shows, there is a strict correlation of the final distribution of distrust values with the initial condition of the network: the range of minimal values for both distrust costs and number of distrusted links is relatively stable, while their maximum values decreases when moving from a configuration that has two sceptic agents as initial nodes to one that has two lazy ones. The result on distrust across the network is less influenced by the role of agents *distributing* the information than by the role of agents *receiving* it.

5.3 Trust, Distrust and consensus

Our last experiment concerns the role of trust and distrust in reaching consensus. As shown in Figure 10, networks with trust *and* distrust present an inverse correlation between size and the number of transmissions that reach consensus: the smaller the network, more often full labelling with a unique formula is obtained (i.e. it is easier to reach consensus). Despite some differences in the reached peaks by lazy and balanced networks, the behaviour is overall similar in all configurations: balanced networks have the highest absolute number of such runs, while networks with higher proportion of sceptic agents have the lowest number of consensus reaching transmissions. Networks with distrust significantly differ from those with trust only for the total amount of consensus-reaching transmissions. We show this for balanced networks in the second graph of Figure 10, the same holding for lazy and sceptic networks: the presence of a distrust routine has a strong impact on the ability of the network to reach consensus in the presence of contradictory information, with no more than 9% of runs reaching a full labelling by either p or $\neg p$ (network of 40 nodes), while in the case of networks with trust only, this value reaches 63% (for networks of the same size).

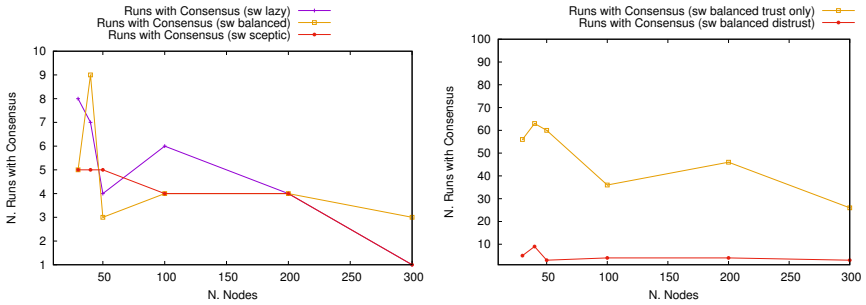


Fig. 10 Consensus in Scale-free Networks with distrust

6 Conclusions

We have presented a logic for the analysis of distrust propagation in a multi-agent system. We have offered related algorithms and an agent-based simulation of the dynamics of such networks when transmitting contradictory information. Our initial experimental results, currently limited to synthetic networks and to be extended with real-world larger data sets, show that: distrust has a lower impact on information transmission in terms of costs than trust; it represents a strong obstacle to reaching consensus; and it qualifies up to a tenth of the size of the network in the presence of contradictory information. Further research will offer extensive comparison with other models, updates of epistemic attitudes and applications to swarm-like phenomena.

References

1. Carbone, M., Nielsen, M., Sassone, V.: A Formal Model for Trust in Dynamic Networks. In: A. Cerone, P. Lindsay (eds.) Int. Conference on Software Engineering and Formal Methods, SEFM 2003., pp. 54–61. IEEE Computer Society (2003). URL http://eprints.soton.ac.uk/262294/. A preliminary version appears as Technical Report BRICS RS-03-4, Aarhus University
2. Chakraborty, S., Ray, I.: TrustBAC: Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems. In: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, SACMAT '06, pp. 49–58. ACM, New York, NY, USA (2006). DOI 10.1145/1133058.1133067. URL http://doi.acm.org/10.1145/1133058.1133067
3. Chandran, S.M., Joshi, J.B.D.: LoT-RBAC: A Location and Time-based RBAC Model. In: Proceedings of the 6th International Conference on Web Information Systems Engineering, WISE'05, pp. 361–375. Springer-Verlag, Berlin, Heidelberg (2005). DOI 10.1007/11581062_27. URL http://dx.doi.org/10.1007/11581062_27
4. Grandi, U., Lorini, E., Perrussel, L.: Propositional Opinion Diffusion. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, pp. 989–997. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2015). URL http://dl.acm.org/citation.cfm?id=2772879.2773278

5. Granovetter, M.: Threshold models of collective behavior. American Journal of Sociology **83**(6), 1420–1443 (1978)
6. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of Trust and Distrust. In: Proceedings of the 13th International Conference on World Wide Web, WWW '04, pp. 403–412. ACM, New York, NY, USA (2004). DOI 10.1145/988672.988727. URL http://doi.acm.org/10.1145/988672.988727
7. Hegselmann, R., Krause, U.: Opinion dynamics and bounded confidence models, analysis, and simulations. Journal of Artificial Societies and Social Simulation **5**(3), 2002
8. Jøsang, A., Pope, S.: Semantic Constraints for Trust Transitivity. In: S. Hartmann, M. Stumptner (eds.) APCCM, *CRPIT*, vol. 43, pp. 59–68. Australian Computer Society (2005)
9. Kempe, D., Kleinberg, J.M., Tardos, E.: Influential nodes in a diffusion model for social networks. In: Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (2005)
10. Lehrer, K., Wagner, C.: Rational Consensus in Science and Society. D. Reidel Publishing Company (1981)
11. Marsh, S., Dibben, M.: Trust, Untrust, Distrust and Mistrust – An Exploration of the Dark(er) Side. In: P. Herrmann, V. Issarny, S. Shiu (eds.) Trust Management, *Lecture Notes in Computer Science*, vol. 3477, pp. 17–33. Springer Berlin Heidelberg (2005). DOI 10.1007/11429760_2. URL http://dx.doi.org/10.1007/11429760_2
12. Massa, P., Avesani, P.: Controversial Users Demand Local Trust Metrics: An Experimental Study on Epinions.com Community. In: Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA, pp. 121–126 (2005). URL http://www.aaai.org/Library/AAAI/2005/aaai05-020.php
13. McKnight, D.H., Chervany, N.L.: Trust and Distrust Definitions: One Bite at a Time. In: R. Falcone, M.P. Singh, Y. Tan (eds.) Trust in Cyber-societies, Integrating the Human and Artificial Perspectives, *Lecture Notes in Computer Science*, vol. 2246, pp. 27–54. Springer (2000). DOI 10.1007/3-540-45547-7_3. URL http://dx.doi.org/10.1007/3-540-45547-7_3
14. Oleshchuk, V.A.: Trust-Aware RBAC. In: I.V. Kotenko, V.A. Skormin (eds.) MMM-ACNS, *Lecture Notes in Computer Science*, vol. 7531, pp. 97–107. Springer (2012). URL http://dblp.uni-trier.de/db/conf/mmmacns/mmmacns2012.html#Oleshchuk12
15. Primiero, G.: A Calculus for Distrust and Mistrust. In: S.M. Habib, J. Vassileva, S. Mauw, M. Mühlhäuser (eds.) Trust Management X - 10th IFIP WG 11.11 International Conference, IFIPTM 2016, Darmstadt, Germany, July 18-22, 2016, Proceedings, *IFIP Advances in Information and Communication Technology*, vol. 473, pp. 183–190. Springer (2016). DOI 10.1007/978-3-319-41354-9_15. URL http://dx.doi.org/10.1007/978-3-319-41354-9_15
16. Primiero, G., Raimondi, F.: A typed natural deduction calculus to reason about secure trust. In: A. Miri, U. Hengartner, N. Huang, A. Jøsang, J. García-Alfaro (eds.) 2014 Twelfth Annual International Conference on Privacy, Security and Trust, Toronto, ON, Canada, July 23-24, 2014, pp. 379–382. IEEE (2014). DOI 10.1109/PST.2014.6890963. URL http://dx.doi.org/10.1109/PST.2014.6890963
17. Primiero, G., Tagliabue, J.: Quantifying epistemic trust in networks with contradictory information. Tech. rep. (2016)
18. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical Reviews E **76**(3) (2007)
19. Wilensky, U.: NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. http://ccl.northwestern.edu/netlogo/ (1999)
20. Zicari, P., Interdonato, R., Perna, D., Tagarelli, A., Greco, S.: Controversy in Trust Networks. In: Procs. 9th Int. Conf. on Trust and Trustworthy Computing (TRUST), Vienna, Austria, August 29-30, 2016, pp. 82–100 (2016). DOI 10.1007/978-3-319-45572-3_5
21. Ziegler, C.N., Lausen, G.: Propagation Models for Trust and Distrust in Social Networks. Information Systems Frontiers **7**(4-5), 337–358 (2005). DOI 10.1007/s10796-005-4807-3. URL http://dx.doi.org/10.1007/s10796-005-4807-3