Distributed problem solving and local greedy updates over a fixed network: theory and empirics.

Atiyab Zafar Fausto Gernone Pia Andres Ariane Donda

November 2025

Abstract

1 Introduction

Decision making under uncertainty is the foremost concern of any organization, from businesses to governments. In practice, individuals solve problems and take decisions under conditions of incomplete knowledge and bounded rationality. Within these limits, what they know—and what they do not—is shaped by their capacity to learn from their environment.

Social networks play a vital role in that learning process (Bala and Goyal 1998; Jackson 2011). By structuring who communicates with whom, and at what cost, the architecture of an organisation can facilitate or hinder the exchange of productive knowledge. The same structure that enables rapid coordination may also constrain the spread of critical insights, with different topologies yielding markedly different outcomes.

A large literature in organizational theory shows that hierarchical architectures can be an efficient response to bounded rationality and costly information processing, by concentrating specialised problem-solving capacity in higher layers while limiting who must communicate with whom (Simon 1962; Radner 1993; Bolton and Dewatripont 1994). At the same time, work on collective problem solving in networks highlights important trade-offs. More integrated, centralised networks speed up the diffusion of successful solutions but risk premature convergence on suboptimal ones, whereas more modular or clustered networks preserve local diversity and can achieve higher long-run performance (Lazer and Friedman 2007; Mason, Jones, and Goldstone 2008; Fang, Lee, and Schilling 2010; Shore, Bernstein, and Lazer 2015; Barkoczi and Galesic 2016).

Some of the network literature examines opinion dynamics, asking when interactions produce consensus or polarisation. The effects of agents' stubbornness, hierarchical network structures, and other forms of heterogeneity are often key features under investigation. The DeGroot model, for example, states that a network will eventually converge to full consensus as long as it is fully connected. In bounded confidence models, consensus and polarisation depend on the extent to which agents trust one another (Noorazar et al. 2020).

The quality of decisions depends on those shared opinions being well-informed. Berekméri and Zafeiris (2020) present a model in which agents can learn both from each other and through observation, with the latter being more costly. Using a genetic algorithm, the authors search over possible network structures and identify those that optimise consensus, accuracy, or a weighted combination of the two. They find that consensus is maximised in highly egalitarian networks in which all participants communicate actively. However, for certain ranges of parameters, accuracy is maximised in more hierarchical networks in which a small number of agents specialise in acquiring and disseminating information, while most agents do not initiate communication. Berekméri and Zafeiris (2020)'s framework is thus a model of information discovery: the task is to learn unknown parameters of the environment as accurately as possible.

Hébert-Dufresne et al. (2025) suggest a model of democratic governance based on a satisfiability problem. They model the policy decisions faced by a population as constrained by logical relations (satisfiable) and represent the structure of information flow as a social hypergraph in which overlapping groups of citizens make coupled decisions. This framework allows them to compare governance regimes ranging from dictatorship to direct democracy, modulating the extent of overlap between voters' groups. They show that intermediate architectures with many small, overlapping decision groups can deliver coherent collective decisions at low coordination cost even in polarised populations.

In this paper, we consider a different type of problem. Rather than estimating an unknown state or parameter vector, agents face a structured problem of constraint satisfaction. Unlike Berekméri and Zafeiris (2020), there is no single scalar "truth" to discover, but there is an underlying set of objective logical relations that determine which combinations of choices are feasible. We are interested in how a fixed communication network shapes a group's ability to identify such internally consistent solutions. Similar to Hébert-Dufresne et al. (2025), our model formalizes this as a Boolean satisfiability task. Agents choose values for a set of binary decision variables, while the environment encodes an unobserved set of Boolean clauses linking these variables. Unlike Hébert-Dufresne et al. (2025), every agent faces the same satisfiability problem and can change her policy choices accordingly. However, knowledge is partial and heterogeneous. As in Berekméri and Zafeiris (2020), agents gradually learn constraints through two channels: direct observation of the environment and communication along the network. Whenever an agent acquires a new clause, it performs local greedy updates of its decision variables to reduce the number of violated constraints, using only the clauses currently in its personal knowledge base. We also allow the environment to drift slowly over time as clauses are replaced, adding dinamicity to the environment and generating a distinction between "new" and "old" information, the latter possibly containing clauses that do not exist any longer.

The theoretical findings are used for an empirical assessment. Using this agent based model[1], we study how different real-life network topologies affect group problem solving. We model networks representing different organizational domains, namely: political blogs, emails between employees of a manufacturing company, and interactions between participants to the 2025 Global Complexity School.[2] These networks are evaluated in terms of the average number of violations across agents, the performance of the best performing agent, and the homogeneity of agents' solutions. Furthermore, across all networks we find a negative correlation between agents' centrality (degree and eigencentrality) and their number of average violations, pointing to the the presence of better connected individuals with better problem solving capabilities. For one of the empirical networks, the communication network of the manufacturing company, we triangulated these results with data on employee managerial level, showing that companies' managers were those characterized by more centrality and better performance. In other words, this model provides a topological characterization of the organization's management roles. Finally, we study how the networks' performance changes by altering their structures - in particular, by changing their inter- and intra-hierarchy connectedness. Results point to an increased performance ....

[Fausto: need to add a comment on the usefulness of comparing networks from very different domains]

2 Methods

2.1 Model

We study how a fixed communication network conditions the rate and quality of individual problem solving when agents face an unknown and evolving set of Boolean constraints. Each agent holds a personal assignment over K binary variables which are interdependent by means of operators AND or XOR (the "clauses" or "constraints")). The agents gradually learns constraints through two channels: private observation and elicitation from neighbours. After acquiring a clause, the agent performs strictly improving local flips on variables appearing in that clause, using only the clauses it currently knows. The design is intentionally lean so that differences in performance can be attributed to network topology and information flow rather than to sophisticated optimisation. We compare different network topologies and parameter values in terms of how well agents perform in terms of:

1. Problem solving: the number of logical constraints agents' opinion vectors violate
2. Homogeneity: the variety between individual agents' opinion vector

2.1.1 Entities and state.

Agents and network. There are N agents $i = 1, \ldots, N$ located on a fixed directed network with weighted ties. Let $w_{ij} \in [0, 1]$ denote the weight on the tie from i to j (absent ties have weight 0).

Decision variables. Each agent i holds a private binary assignment

$$x^{(i)} = \left(x_1^{(i)}, \ldots, x_K^{(i)}\right) \in \{0, 1\}^K.$$

Universal constraints. At time t, the environment contains a hidden set of M clauses

$$\mathcal{C} = \{c_1, \ldots, c_M\}, \qquad M = \mathrm{round}(\alpha K).$$

[1] An interactive web application can be accessed at `https://cgs2025.shinyapps.io/networked-problem-solving/` which runs the model of your choice. See Appendix C for more.

[2] A wider selection of networks is reviewed in the Appendix.

Each clause c involves either two or three distinct variable indices related by means of operators of the type AND and XOR. For each individual, a certain clause is satisfied if her choices of decision variables satisfy the clause's constraints.

Knowledge bases. Agent i maintains a personal knowledge base $\mathcal{C}_i \subseteq \mathcal{C}$ that records clauses it has learned via observation or communication. The implementation keeps $|\mathcal{C}_i| \leq M$ by discarding the oldest entries when necessary and removes duplicates after each learning step.

Violation counts and homogeneity. For each agent i, define the true violation count as

$$V_i^{\text{true}} \;=\; \big|\{\, c \in \mathcal{C} \;:\; x^{(i)} \text{ violates } c \,\}\big|.$$

We report the population average $\bar{V}^{\text{true}}$ and the best performance $V_{\min}^{\text{true}} \;=\; \min_i V_i^{\text{true}}$. Tracking both is useful because some organisations rely on a few well-informed individuals to make high quality decisions, whereas in other settings performance depends on most agents making few mistakes.

Finally, we characterise the homogeneity in agents' individual solutions. For each variable x_a, we calculate the share of agents holding the majority value, and average this over all K variables. The homogeneity range is $H \in [0.5, 1]$. Higher H indicates that agents tend to agree on each variable. Whether this is beneficial depends on context: high agreement can reflect convergence on good decisions or premature alignment on poor ones.

2.1.2 Initialisation

The initial state is consistent with the objects defined above.

1. **Constraint set.** Draw M clauses. For each clause c, sample ....
2. **Communication network.** Load a dataset graph.
3. **Beliefs and knowledge.** Initialise random beliefs $x^{(i)}(0) \sim$ i.i.d. Bernoulli$(0.5)^K$ for all i, and set $\mathcal{C}_i(0) = \emptyset$. At baseline, agents know no constraints and therefore cannot improve except when they begin to learn them.

2.1.3 Per period dynamics

Time proceeds in discrete periods $t = 1, 2, \ldots, R$. Within each period, all agents update in parallel according to four steps. The learning steps add clauses to $\mathcal{C}_i$; the local update steps attempt flips that strictly reduce the number of violations relative to the subset of $\mathcal{C}_i$ that is relevant to the clause just learned.

Step 1: private observation. With probability $p_{\text{obs}} \in [0, 1]$, agent i samples a clause c uniformly from $\mathcal{C}(t)$ and updates $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{c\}$. Duplicates are harmless (they are removed) and only slow learning if present.

Step 2: local greedy update around the observed clause. ...

Step 3: neighbour elicitation. Agent i selects neighbour j according to $\mathbb{P}(i \to j)$. If j knows at least one clause that i does not, draw one uniformly from $\mathcal{C}_j \setminus \mathcal{C}_i$, call it c', and set $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{c'\}$. This step captures directed, weighted information flow over the network.

Step 4: local greedy update around the received clause. Repeat the same one-by-one update as in Step 2 with clause c' in place of c. In words, whenever new information about relevant variables arrives, the agent immediately attempts strictly improving local repairs.

At the end of each period, performance measures (i.e., violations and homogeneity) are reported.

2.1.4 Environmental change

To introduce drift, every τ_{clause} periods one clause in the universal set is replaced:

$$\text{draw } u \sim \text{Unif}\{1, \ldots, M\}, \qquad c_u \leftarrow c^{\text{new}},$$

where c^{new} is drawn using the same procedure as at initialisation. Agents do not automatically forget clauses, so $\mathcal{C}_i(t)$ may contain outdated information until it is superseded by further learning. This creates a natural tension between fast diffusion of useful constraints and the propagation of stale ones.

2.1.5 Termination

Simulations run for a fixed horizon R periods.

2.2 Network decomposability and performance

The study of real-world networks begs the question of whether it is possible to improve their performance by altering their topology. In particular, we focus on the impact of network clusters on its performance. Highly clustered, almost decomposable architectures can support parallel local adaptation and protect exploratory subgroups from premature convergence (Simon 1962; Fang, Lee, and Schilling 2010), but if communities are too insulated, useful constraints discovered in one block will rarely diffuse to others.

We study the effect of communities or clusters through a rewiring exercise.[3] In particular, we use a switching algorithm or local rewiring algorithm (LRA) which preserves the degree of the network but rewires it. The normal LRA algorithm picks up a random pair of edges. Say, (a, b) and (c, d). It then picks up the leading or the sub-leading pair of nodes and exchanges them to (a, d) and (c, b). This ensures that every time we rewire the in-degree of d and b as well as the out-degree of a and c remains same.

Our randomisation method is illustrated in Figure 1. We first use a graph theoretic algorithm to detect communities. Once we have the communities, we pick up a fraction of intra community links (i.e., links between nodes that belong to same community) and we remove it. Instead of that, we create an inter community link. Thus making sure the number of links remain same and the links to node ratio also remains same.

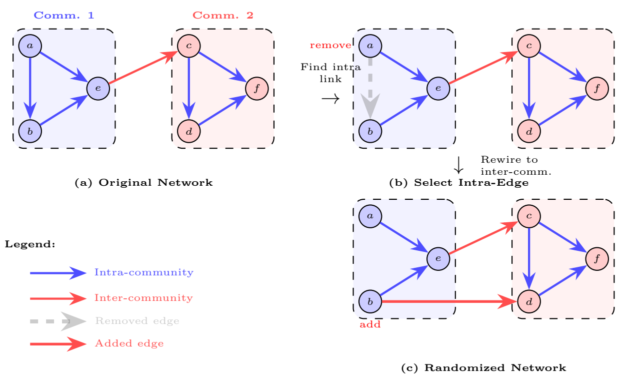


Figure 1: Randomisation Algorithm

3 Network Data

We run our model on the following empirical networks.

Congress Twitter. Fink et al. (2023) construct a weighted, directed network dataset based on the Twitter activity of members of the US Congress between February and June 2022. Edges are weighted by what the authors call 'probabilities of influence'. The probability of influence on a member of Congress j from another member i is calculated as the number of retweets, quote tweets, replies or mentions of member i's tweets by member j, divided by the total number of times member i tweeted. In other words, the weights refer to the percentage of tweets of one member which another member publicly reacted to.

Company Email Exchanges. This dataset underlies part of the analysis in Michalski, Palus, and Kazienko (2011), and captures the e-mail communication network of a mid-sized manufacturing company. Each node represents an e-mail address, and each directed edge an e-mail sent from one e-mail address to another.

Political Blogs. Developed by Adamic and Glance (2005), this network includes 1,489 US political blogs. The network is directed, with each edge representing a hyperlink from one blog to another. Only the main component of the network was included.

[3] There are various ways randomisations of a network can be done. See Clerck, Utterbeeck, and Rocha (2024) for a review on various methods.

Conference Attendance. This data describes the collaborations emerging from an in-person conference. The data was made available courtesy of Research Corporation for Science Advancement (RCSA). RCSA organises conferences in their *scialog* (science dialog) program. It brings together some early career researchers as well as a handful of experienced scientists together. Throughout the conference the people interact and are told to work together. At the end the researchers submit collaborative research proposals. Each person can be in maximum two proposals. And out of 25-30 only 5 to 10 are selected for funding. We have created collaboration network from the conferences, where each node is an individual and each edge implies that the pair collaborated together.

Complexity Global School Interactions. This network was created using anonymised data from the 2025 Complexity Global School Arch server. If a user messaged another user, a link was created from the first user to the second user. This messages could be either to students personal account, or to a faculty. If a user talked with another user inside a space (a discussion group or a project group we connect them by a link) and finally we also connect the users if they were writing on the same etherpad. This takes care of collaborations between project groups.

The datasets are summarised in Table 1.

Table 1: Summary of network datasets

Network Name	**Source**	**Nodes**	**Edges**
Congress Twitter	Fink et al. (2023)	Members of US Congress (2022)	Directed, weighted by probability of influence; calculated as reactions (retweets, quotes, replies, mentions) to a member's tweets divided by their total tweets
Company Email Exchanges	Michalski, Palus, and Kazienko (2011)	Company employees	Directed, unweighted; edge represents an email sent
Political Blogs	Adamic and Glance (2005)	1,489 US political blogs	Directed, unweighted; edge represents a hyperlink from one blog to another
Conference Attendance	Research Corporation for Science Advancement (RCSA)	Conference attendants	Undirected, unweighted; an edge represents a collaboration between attendants
Complexity Global School Interactions	ARCH platform	Attendants to CGS	Mixture of interactions: online messages, interaction on notepad function, etc..

4 Results

This section presents the results. The model was run with $p_{\text{obs}} = 0.03$, so that in a network of 100 agents, on average 3 agents observe a new clause from the environment in each period. Keeping the observation probability relatively low increases the relative importance of the communication network as a channel for information acquisition, thereby highlighting structural differences between networks.

Secondly, τ_{clause} was set at 10, so that every 10 periods one clause is selected at random and altered. The qualitative results were stable across different values of τ_{clause}, with more dynamic environments simply widening the error range by increasing the stochastic component of the model.

Finally, the results presented below show series of 1,000 periods: a sufficient number of iterations for the key patterns to emerge and for the results to stabilise.

4.1 Networks performance

In table 2 we have listed various network parameters for each network that we have studied, while Figure 2 compares the evolution of violations and homogeneity across the empirical networks. Marked differences across networks emerge.

Table 2: Key Network Properties and Performance Metrics. Key: N = nodes; E = edges; D = edge density (percentage of possible directed edges present, excluding self-loops); C_D = degree centralization; f_{inter} = inter-community edge fraction; ρ = reciprocity; CC = clustering coefficient; V = baseline violations (averaged over 50 seeds); H = baseline homogeneity. Networks listed in order of appearance in dataset. The Violations and homogeneity are calculated after 500 graph updates.

Network	N	E	D **(%)**	C_D(%)	f_{inter}	ρ	CC	V	H
Congress Twitter	475	13,289	5.9	21.64	0.208	0.462	0.224	35.4	0.812
Company Emails	167	5,784	20.9	41.77	0.658	0.876	0.559	38.6	0.777
Political Blogs	1,222	19,024	1.3	15.28	0.074	0.243	0.219	44.0	0.699
Conference Collab.	63	224	11.5	9.1	0.241	1	0.295	36.0	0.80
CGS Interactions	64	180	2.8	11.8	0.144	0.256	0.406	48.8	0.646

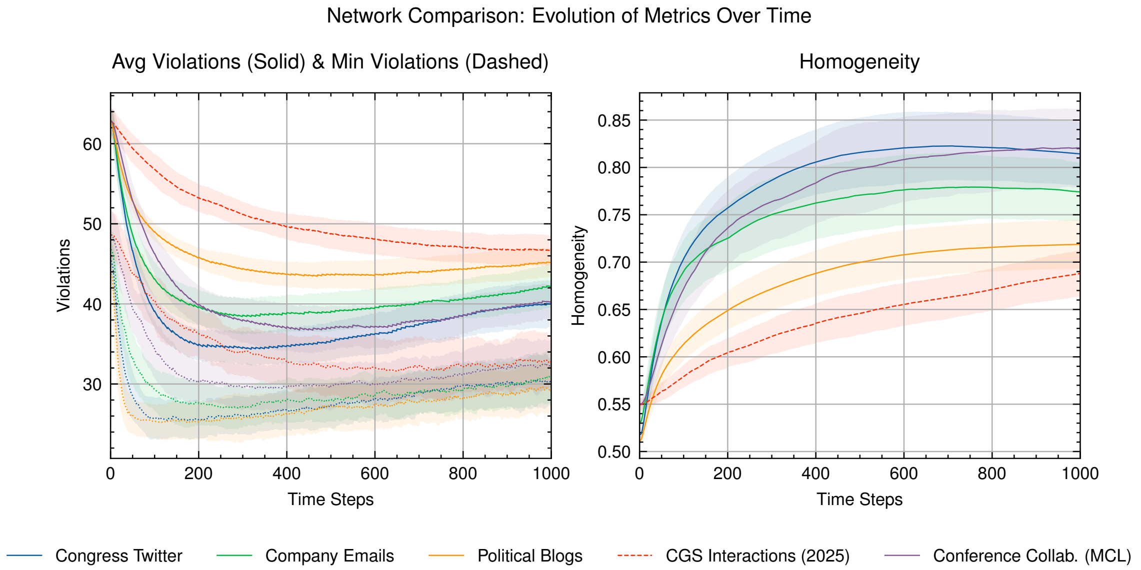


Figure 2: The performance of the model on various empirical networks showing a range of variation across average violations as well as homogeneity. These plots are average for 50 different runs with the band representing 1 standard deviation from the mean across all runs. The dashed curves are CGS networks and MCL is a conference network. See appendix for more detail.

In terms of average violations, the MCL conference network, Congress Twitter and the company email network converge to a similar level in the long run, while Political Blogs and CGS stabilise at higher values. The three best performing networks are heterogeneous in terms of size and centrality, but they are all characterised by relatively high edge density (11.5%, 5.9% and 20.9%) and by many reciprocal ties (100%, 46% and 88%). This pattern points to the importance of frequent, mutual communication for the spread of new information and for keeping average violations low. Interestingly, while the Congress Twitter network is considerably less dense and less reciprocal than the other two, it still performs remarkably well, especially in the long run. This can be interpreted as an advantage conferred by its sheer size, roughly three times that of the email network and almost eight times that of the MCL conference. The same interpretation, however, begs the question of why the Political Blogs network, with $N = 1{,}222$ nodes and by far the largest, performs so poorly. Its peculiar peanut shaped topology offers a partial explanation: the network is effectively split into two communities with very little communication between them, so new clauses discovered on one side are rarely communicated to the other, leaving many agents with a high number of violations.

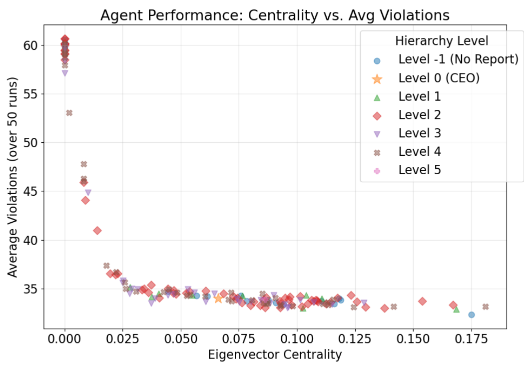
Looking at minimum violations, the ordering is almost reversed. The Political Blogs network consistently yields the lowest minimum violation, followed by the company emails and Congress Twitter, with MCL and especially CGS performing worse. This is largely a sample size effect: minima are taken over all agents in a network, and the Political Blogs network is by far the largest ($N = 1{,}222$), followed by Congress ($N = 475$) and the email network ($N = 167$). Larger networks have a greater chance of containing at least one exceptionally well connected and well informed agent whose individual assignment fits almost all constraints. The smaller MCL and CGS networks (both around 60 nodes) simply have fewer draws from the underlying performance distribution, so their best performers remain noticeably worse even when the average level of violations is comparable. Following this interpretation, the relatively strong performance of the email network, almost on a par with Political Blogs despite being less than 14% of its size, remains to be explained. The efficiency of the best performers within the company network could be the result of its highly centralised structure which, with a degree centralisation of about 42%, almost resembles a hub and spoke topology. Such a concentrated structure is arguably capable of producing “stars”: agents whose central position allows them to make the most of the new information observed elsewhere in the company and to appropriate it quickly.

Finally, homogeneity shows a different ranking again. The Congress Twitter network achieves the highest homogeneity throughout most of the run, with the MCL conference close behind and catching up only near the end of the simulation. The company email network takes third place, with Political Blogs and CGS following. Intuitively, the two factors that appear to influence the extent to which agents propose the same solution to the satisfiability problem are the density of the network and its centralisation. High density increases agents' communication and, more specifically, correlates the order in which new clauses are discovered across agents, making it more likely that greedy local updates deliver similar outcomes. The company email network, however, only takes third place despite being by far the most dense network. This is likely due to the fact that it is also the most centralised, so that its topology is highly asymmetric, leading to differences in the timing and, probably, the order of information received by different nodes. As a result, the solutions adopted by the various nodes may diverge.

4.2 Node heterogeneity

In this section we consider the effects of our model locally at the node level. In Figure 3a we plot the centrality and violations averaged over 50 runs for the company email network. Centrality of course does not change over the run. We notice a decay relation ship between a node's centrality and its violations. There is a steep drop off at low centrality. That is, agents with very low centrality (near zero) have dramatically higher violations. While agents with even modest centrality converge to a stable low violation. The nodes with subsequently higher centrality flattens out to show a plateau behavior. These two are characteristic of power law relationships.

The email network dataset comes with the information of which person reported to whom in the manufacturing company (Nurek and Michalski (2020)). The highest ranked person in the company is the CEO (node 86) and everyone else works under him at different levels defined from one to 5. As we have the node level data and hierarchies we can study how agents who work at different levels in the company perform. This analysis and the details of how the levels were calculated can be obtained from the Appendix B.2.



(a)

Figure 3: Local analysis for the email network of the manufacturing company. 3a show the plot for Average Violations for 50 runs of each node v/s the degree centrality of that node. 10a shows the statistics of node centralities (in-degree) for each hierarchical level of the network, and 10b shows the statistics for violations for nodes at different levels.

4.3 Bridging clusters

We study the effect of rewiring (and consequently decreasing the insularity of communities) on one specific network: that representing political blogs n the US. This network was famously introduced in an article named ”divided they blog” (Adamic and Glance 2005), pointing to a peculiar, peanut-shaped topology whereby the network is divided in two main communities or clusters: one leaning towards the democrats and the other leaning towards the republicans. Blogs that tend to be politically aligned towards one side tend to cite the blogs that are similar to them politically. This creates a peanut-like network, as shown in Figure 4, where the intra-community or intra-hierrarchical links are way more than inter.

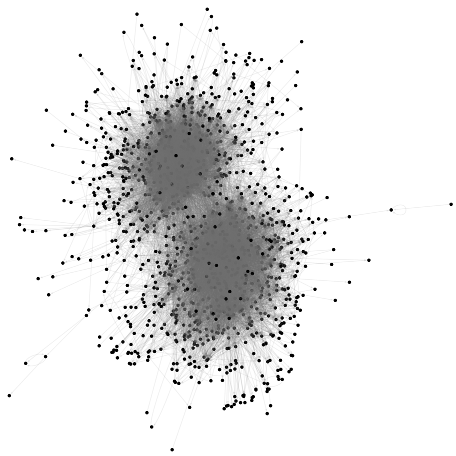


Figure 4: Network visualisation of political blogs network, drawn using Graphviz Gansner and North 2000

We created individual randomised networks for the political blog networks and found the following results:

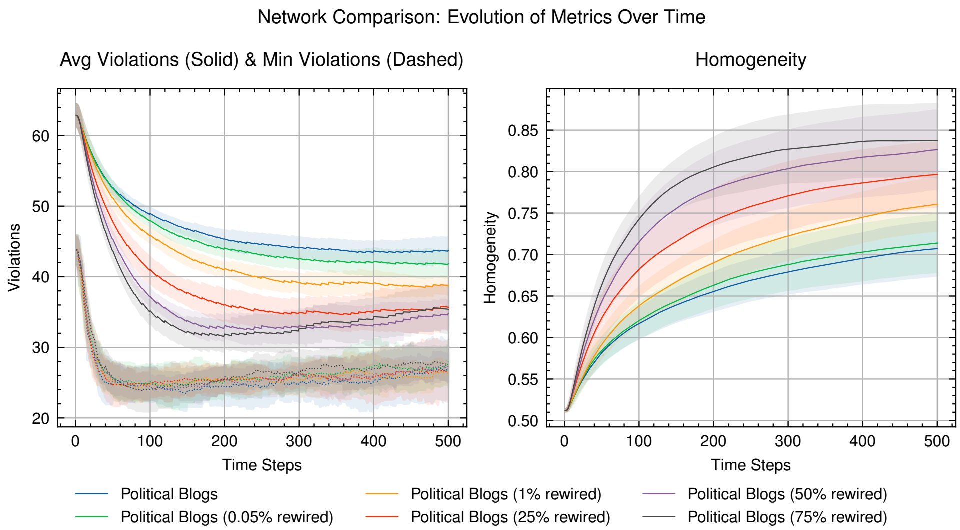


Figure 5: Model results for political blogs network and its randomisations to create more vertical connectivity. Average for 10 runs for each network

As we increase the inter community links and consequently reduce the polarisation of the network, performance improves in terms of both average violations and homogeneity. Remarkably, as little as a 0.05% rewiring (corresponding to approximately 10 out of 19,024 edges) generates sizable gains, with diminishing but still positive marginal improvements for much larger rewiring. For average violations, the apparent optimum is around 25% rewiring, beyond which additional rewiring does not significantly affect performance. This occurs because clauses that are discovered in one cluster can more easily flow to the other, although this benefit wanes once a threshold of average path length is reached. For homogeneity, the benefits of rewiring continue all the way to a full network rewiring, because the synchrony of agents' discoveries keeps increasing with connectivity across communities: as more cross community ties are added, agents tend to receive new clauses in a more similar temporal order and greedy local updates are more likely to align.

5 Conclusion

6 Author Contributions

7 Acknowledgment

The authors would like to thank all the faculty at Complexity Global School 2025 held at Bogota, Colombia organised by the Santa Fe Institute . The discussion that happened during the school was significant in development of this project. Discussion with Joseph Bak-Coleman in Bogota helped a lot in performing literature survey.

The conference dataset was provided by Research Corporation for Science Advancement(RCSA) via Emma Zajdela and Joan Kim at Santa Fe institute helped in acquiring of the Arch data for the school.

Finally, the guidance of our mentors Juan Restrepo, Emma Zajdela and our peer mentor Lavanya Kosuri was really important in shaping up of this paper.

References

Adamic, Lada A and Natalie Glance (2005). "The political blogosphere and the 2004 US election: divided they blog". In: *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43.

Bala, Venkatesh and Sanjeev Goyal (1998). "Learning from Neighbours". In: *Review of Economic Studies* 65.3, pp. 595–621.

Barkoczi, Daniel and Mirta Galesic (2016). "Social Learning Strategies Modify the Effect of Network Structure on Group Performance". In: *Nature Communications* 7, p. 13109. DOI: 10.1038/ncomms13109.

Berekméri, Evelin and Anna Zafeiris (2020). "Optimal collective decision making: Consensus, accuracy and the effects of limited access to information". In: *Scientific Reports* 10.1, p. 16997.

Bolton, Patrick and Mathias Dewatripont (1994). "The Firm as a Communication Network". In: *The Quarterly Journal of Economics* 109.4, pp. 809–839.

Clerck, Bart De, Filip Van Utterbeeck, and Luis E. C. Rocha (2024). *Comparative analysis of graph randomization: Tools,methods, pitfalls, and best practices.* arXiv: 2405.05400 [physics.soc-ph]. URL: https://arxiv.org/abs/2405.05400.

Fang, Christina, Jeho Lee, and Melissa A. Schilling (2010). "Balancing Exploration and Exploitation through Structural Design: The Isolation of Subgroups and Organizational Learning". In: *Organization Science* 21.3, pp. 625–642. DOI: 10.1287/orsc.1090.0468.

Fink, Christian G et al. (2023). "A congressional twitter network dataset quantifying pairwise probability of influence". In: *Data in Brief* 50, p. 109521.

Gansner, Emden R and Stephen C North (2000). "An open graph visualization system and its applications to software engineering". In: *Software: practice and experience* 30.11, pp. 1203–1233.

Hébert-Dufresne, Laurent et al. (2025). "Governance as a complex, networked, democratic, satisfiability problem". In: *npj Complexity* 2.1, p. 14.

Jackson, Matthew O. (2011). "An Overview of Social Networks and Economic Applications". In: *Handbook of Social Economics*. Ed. by Jess Benhabib, Alberto Bisin, and Matthew O. Jackson. Vol. 1. North-Holland, pp. 511–585.

Lazer, David and Allan Friedman (2007). "The Network Structure of Exploration and Exploitation". In: *Administrative Science Quarterly* 52.4, pp. 667–694.

Mason, Winter A., Andrew Jones, and Robert L. Goldstone (2008). "Propagation of Innovations in Networked Groups". In: *Journal of Experimental Psychology: General* 137.3, pp. 422–433.

Michalski, Radosław, Sebastian Palus, and Przemysław Kazienko (2011). "Matching organizational structure and social network extracted from email communication". In: *International conference on business information systems*. Springer, pp. 197–206.

Noorazar, Hossein et al. (2020). "From classical to modern opinion dynamics". In: *International Journal of Modern Physics C* 31.07, p. 2050101.

Nurek, Mateusz and Radosław Michalski (2020). "Combining Machine Learning and Social Network Analysis to Reveal the Organizational Structures". In: *Applied Sciences* 10.5, p. 1699.

Radner, Roy (1993). "The Organization of Decentralized Information Processing". In: *Econometrica* 61.5, pp. 1109–1146.

Shore, Jesse, Ethan S. Bernstein, and David Lazer (2015). "Facts and Figuring: An Experimental Investigation of Network Structure and Performance in Information and Solution Spaces". In: *Organization Science* 26.5, pp. 1432–1446.

Simon, Herbert A. (1962). "The Architecture of Complexity". In: *Proceedings of the American Philosophical Society* 106.6, pp. 467–482.

A Random Networks

A.1 Comparison of Different randomisations

The randomisation algorithm discussed in the section 2.2 provides better performance change compared to the standard rewiring algorithm. In Figure 6, we plot network performance measures using both rewiring algorithm.

In both cases the same proportion of links were rewired (10%). In the standard algorithm we picked up any random link whereas in the other algorithm we change an intra-community link to inter-community link.

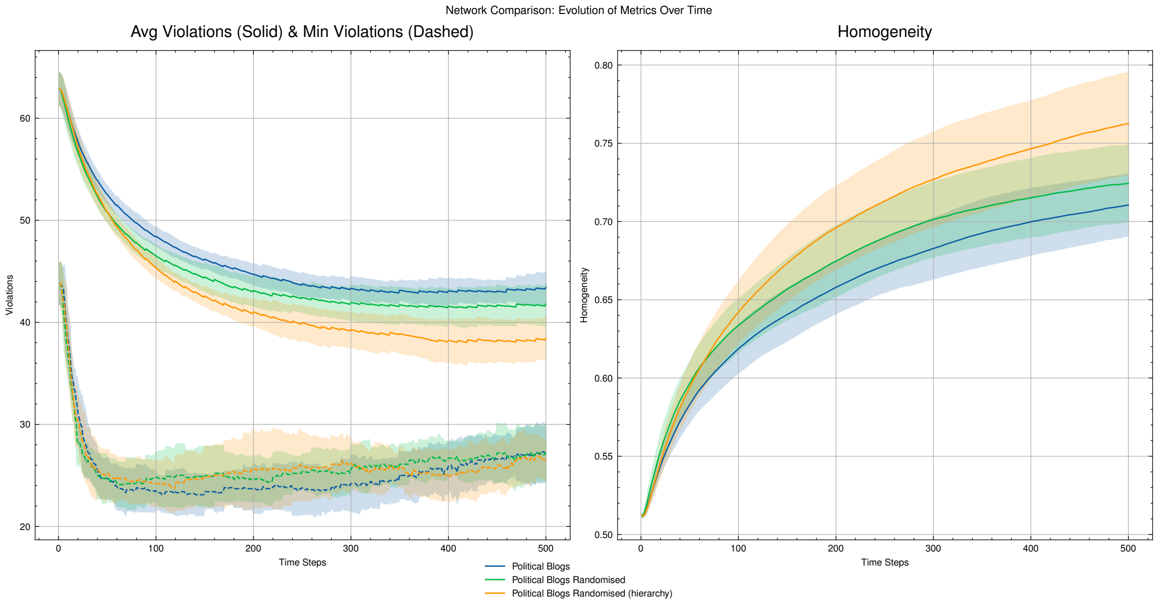


Figure 6: Political Blogs Network performs as good as congress twitter network after randomisation. Our algoriothm gives better results compared to the standard rewiring.

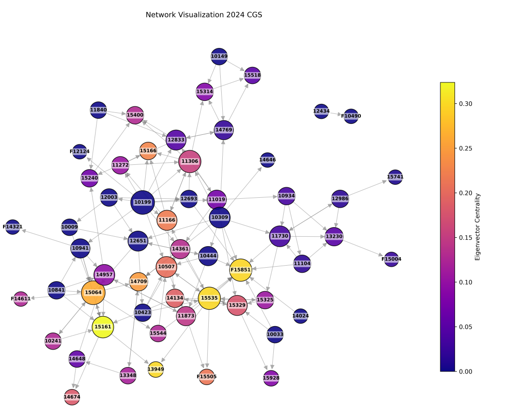
In the standard rewiring rule, represented by the green line, there is a slight improvement in performance, ~ 0.71 to ~ 0.75 for homogeneity and a drop of ~ 2 in the average violations seen across the network. The hierarchical rewiring (orange line) performs best, achieving lowest average violations ($\sim 38-39$) and highest homogeneity ($\sim 0.77-0.78$). This demonstrates that strategically increasing cross-community connections while preserving local structure enhances information diffusion.

B More on Empirical Networks

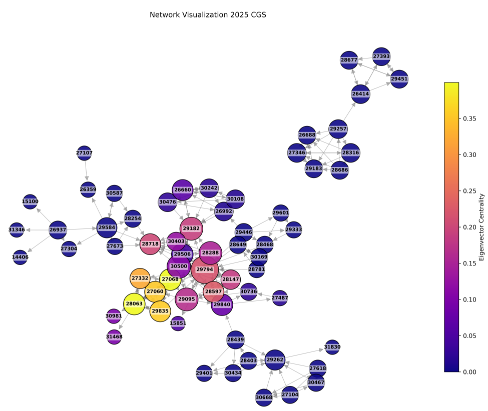
B.1 Conference and ARCH data

B.1.1 ARCH Networks

Figure 7 shows the network of users interacting at the Arch platform shared by the Santa Fe institute. The nodes size are scaled by the degree of the node whereas the nodes are colored by the eigenvector centrality.



(a) CGS 2024 Network using Interactions from Arch platform



(b) CGS 2025 Network using Interactions from Arch platform

Figure 7

B.1.2 Conference Data

Data of four in-person conference (MCL, TDA, CMC, AES). We have created collaboration network from the conferences, where each node is an individual and each edge implies that the pair collaborated together. The data was made available courtesy of Research Corporation for Science Advancement(RCSA). RCSA organises conferences in their *scialog* (science dialog) program. It brings together some early career researchers as well as a handful of experienced scientists together. Throughout the conference the people interact and are told to work together. At the end the researchers submit collaborative research proposals. Each person can be in maximum two proposals. And out of 25-30 only 5 to 10 are selected for funding.

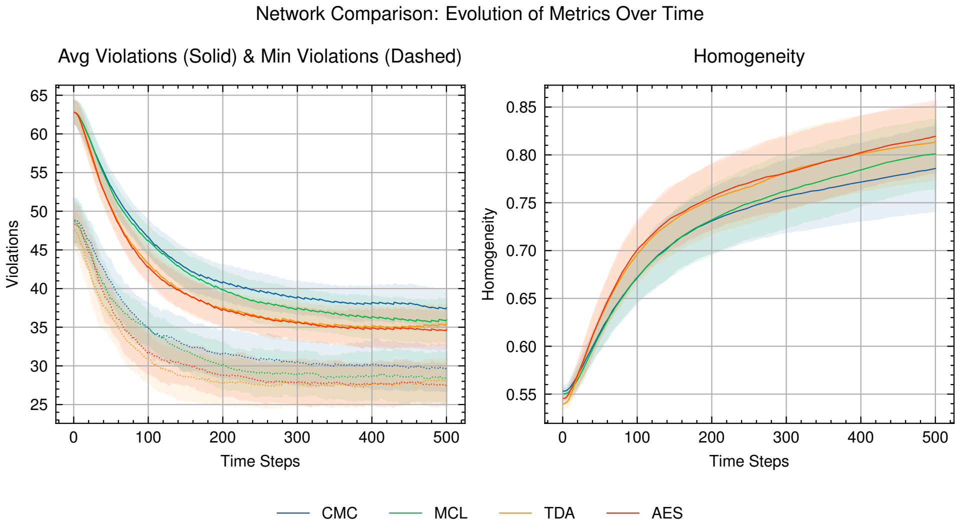


Figure 8: Model's performance from colaboration network from 4 Conferences

B.2 Company Email Network

B.2.1 Hierarchy of the company

Hierarchy in the dataset represents the "chain of command" or "who reports to whom." It's like a family tree for a company. Here is how we calculated it from the data provided . In the dataset we were given two pieces of information for every employee: their ID (agent's unique ID) and their manager's ID (who is their boss). The person at top was mentioned in the information of the dataset as the CEO of the company which we define as level 0. Once we have the CEO we calculate everyone's rank by counting how many steps away they are from the CEO. Thus, Level 1 is all agents who report directly to the CEO and agents at level 2 report to a Level 1 Manager and so on. This is done using a shortest path algorithm in Networkx. See Figure 9 for a visualisation.

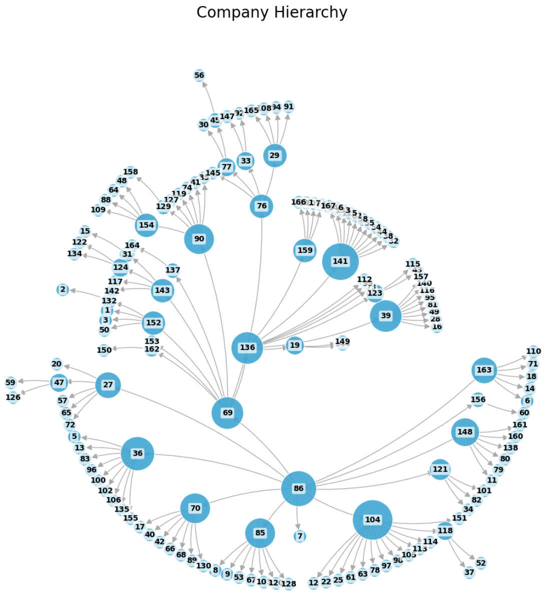
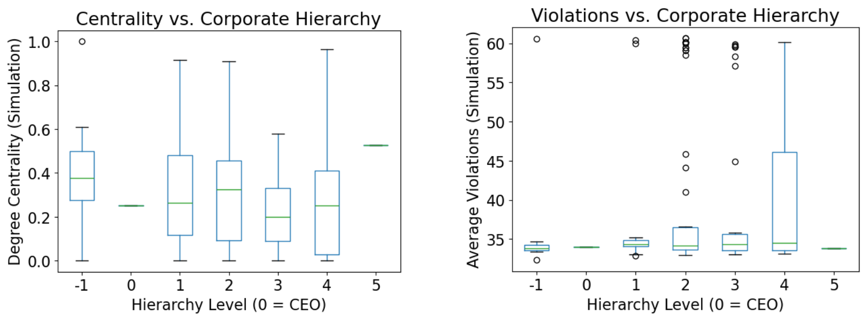


Figure 9: Network representing Company Hierarchy obtained from the data. Node size are scaled by the degree of the node.

There are unconnected nodes, we label them as Level -1 or undefined to separate them from the main corporate hierarchy. There are 13 of such accounts. 7 of these are listed as 'technical email account - not used by employees' and 6 of them are 'former employee account'.

B.2.2 Statistics of hierarchal levels



(a) (b)

Figure 10: Figure 10a shows the statistics of node centralities (in-degree) for each hierarchical level of the network, and 10b shows the statistics for violations for nodes at different levels

We study in Figures 10a and 10b different statistics of centrality and violations. There is only one node at the top (level 0) with low degree centrality as well as very low average violation, i.e. very high efficiency. This is the CEO (shown as Star in Figure 3a). Level 1 (upper management) are the direct reports to the CEO. The box is small with median close to the CEO pointing to high efficiency but there are a few outliers with violation (~ 60) indicating that there are a few direct reports that are not performing as well despite their high rank.

Middle management roles have higher spread but interestingly the best performers even at this level are as good as the CEO. The number of outliers are even higher in this case, which suggests that "many middle managers" are not getting enough information.

Level 4 is the worst performing layer. The box which shows 75th percentile goes as far as 45 and the upper whisker is as bad as the outliers (maximum violators). Although the median for each level is at low violation, the distribution varies a lot and the average violation keeps getting worse farther we are from the center. Nodes at -1 Level are "informal" nodes, these include emails that are just used for technical reasons or include retired individuals. Tables 3 and 4provides table of people and their hierarchies of the best performing nodes and worst performing nodes.

Agent_ID	**Centrality**	**Avg_Viol**	**Level**
0	1.000000	32.32	-1
36	0.915790	32.88	1
17	0.713311	33.00	2
85	0.505724	33.02	1
42	0.407336	33.04	2

Table 3: Best Performing Agents

Agent_ID	**Centrality**	**Avg_Viol**	**Level**
161	0.0	60.66	2
155	0.0	60.66	2
139	0.0	60.56	-1
163	0.0	60.44	1
143	0.0	60.14	2

Table 4: Worst Performing Agents

C App

We have developed an interactive application that allows users to explore our agent-based model in real time. The app features a network visualization panel showing the communcation graph as an interactive plot. Users can change various parameters of the network and setup their own environment then let the simulation run. The plots for tracking violations and homogeneity can be found in tabs called "Performance" and "Distribution".

Users can run simulations on multiple real-world datasets (Political Blogs, Congress Twitter, Company Emails) and observe how network structure influences collective problem-solving performance. Users can also upload their own network as a `graphml` file and see the model evolve with time. and The app is accessible at `https://cgs2025.shinyapps.io/networked-problem-solving/` and serves as an educational tool for understanding the interplay between social networks and distributed cognition.

We are using Shiny (Python) for front end and NetworkX, Mesa (agent-based-modeling) in the backend. The app is optimised to work best on bigger screen devices but will still work on smartphones.