Architectures of Problem Solving: How Network Structures Shape Decision Making

Atiyab Zafar* Fausto Gernone† Pia Andres‡ Ariane Donda§

November 2025

Abstract

Individuals and organizations are continually tasked with complex decisions under incomplete information. We present an agent-based model of distributed problem solving in which individuals face a shared Boolean satisfiability task. Agents hold only partial knowledge of the underlying logical constraints and update their binary policy choices through local greedy repairs whenever they acquire new clauses, either from direct observation or communication within a fixed network. Because behavioral assumptions are intentionally simple and homogeneous, all systematic variation in collective performance arises from network structure.

Using this framework, we evaluate several real-world communication networks – including Twitter interactions between legislators and corporate email exchanges – on their ability to support accurate and coherent problem solving. Dense, reciprocal, and well-integrated networks converge to low violation counts and high homogeneity, whereas sparse or community-segmented networks trap information locally, hindering the discovery of consistent solutions. Node-level patterns reveal strong performance advantages for central agents, which we validate using managerial data from a manufacturing firm. Finally, rewiring experiments show that modest increases in inter-community connectivity can yield substantial performance gains. Taken together, these results highlight general design principles linking network topology to collective intelligence.

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 organization 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 specialized 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, centralized 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 polarization. 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 polarization 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 optimize consensus, accuracy, or a weighted combination of the two. They find that consensus is maximized

*Department of Physics, University of Delhi.
†University College London. Corresponding author: fausto.gernone.20@ucl.ac.uk
‡Durham University & London School of Economics
§Centro para Imaginação Crítica

in highly egalitarian networks in which all participants communicate actively. However, for certain ranges of parameters, accuracy is maximized in more hierarchical networks in which a small number of agents specialize 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 polarized 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 dynamicity to the environment and generating a distinction between 'new' and 'old' information, the latter possibly containing clauses that no longer exist.

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: Twitter interactions between members of the US Congress; email exchanges between employees of a manufacturing company; US political blogs; collaborations between attendees of the same scientific conference; and interactions between participants to the 2025 Complexity Global 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. By examining networks drawn from a wide range of organisational settings, we hope to identify patterns and relationships between topology and collective intelligence that transcend specific contexts.

Across all networks, we find a negative correlation between agents' centrality (degree and eigencentrality) and their average number of violations, pointing to the presence of better connected individuals with better problem solving capabilities. We illustrate this result with reference to one specific empirical network: the communication network of the manufacturing company. Finally, we study how the networks' performance changes by altering their structures - in particular, by altering the extent to which the network communities are connected with one another. Results point to an increased performance through rewiring at low levels of inter-community connectivity, with homogeneity improvements continuing steadily at higher levels too. This is driven by higher synchronicity in agents' information updates and shows how polarization can result simply from the order in which information is received by otherwise identical agents.

The remainder of this paper proceeds as follows. Section 2 presents our model and computational approach. Section 3 provides an overview of the datasets used in the empirical analysis. Section 4 presents and interprets the results. Section 5 concludes.

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 agent gradually learns constraints through two channels: private observation and elicitation from neighbors.

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

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

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 optimization. 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; and
2. Homogeneity: the variety between individual agents' opinion vector.

2.1.1 Entities and state.

Agents and network. There are N agents $i = 1, \dots, N$ 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

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

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

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

Each clause c involves two distinct variable indices related by means of operators of the type AND and XOR. For each agent, a certain clause is satisfied if its choice of decision variables satisfies the clause's constraints. α is a real number determining the density of clauses per decision variable.

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

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

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 organizations 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 characterize 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 use the same solutions to the set of constraints. Whether this is beneficial depends on context: high agreement can reflect convergence on good decisions or premature alignment on poor ones.

2.1.2 Initialization

The initial state is consistent with the objects defined above.

1. **Constraint set.** Draw M clauses randomly.
2. **Communication network.** Load a dataset graph.
3. **Beliefs and knowledge.** Initialize 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 $t = 1, 2, \dots, R$. 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. Each agent knowledge base is capped to a maximum of M clauses: once the cap is reached, as new information is learned, old clauses are dropped.

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 $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{c\}$.

Step 2: local greedy update around the observed clause. Upon learning a clause c, the agent tries flipping each of the variables included in c if and only if the flip strictly reduces the number of violations in the subset of C_i that is relevant to those variables.

Step 3: neighbor elicitation. In each period, agent i may elicit information from its neighbors. The chance that i manages to elicit a clause from others is proportional to how many inbound connections it has (or, in weighted networks, to the total weight of those inbound links). We rescale the edge weights so that, on average across the whole network, there is one elicited clause per agent and period, with highly connected agents will typically eliciting more clauses than poorly connected ones. Conditional on an elicitation event from neighbout j to agent j; if j knows at least one clause that i does not, we draw one uniformly from $\mathcal{C}_j \setminus \mathcal{C}_i$, call it c', and update $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{c'\}$. This step captures directed, weighted information flow over the network, with more central agents eliciting much more information than peripheral ones.

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 initialization. Agents do not automatically forget clauses, so $\mathcal{C}_i(t)$ may contain outdated information until it is superseded by further learning. As a result, agents' performance is determined not by the size of their knowledge base, but by how updated their information is.

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 randomization method is illustrated in Figure 1. We first use a graph theoretic algorithm to detect communities. We then pick up a fraction of intra community links (i.e., links between nodes that belong to same community), remove them, and replace them with inter community links. This ensures that the number of links and the links to node ratio both remain unchanged.

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

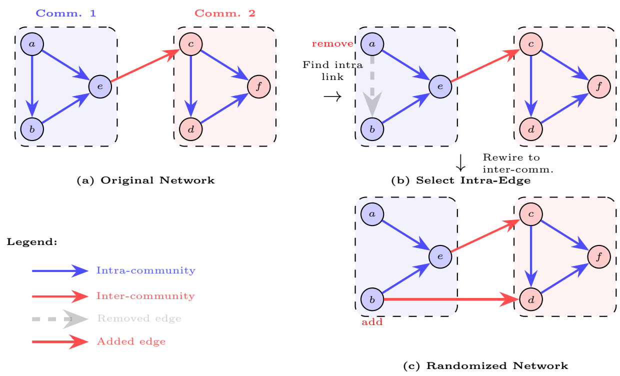


Figure 1: Randomization Algorithm

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. Appendix B.3 provides more detail on this dataset.

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.

Conference Attendance. This data describes the collaborations emerging from the 2015 edition of the in-person conference 'Molecules Come to Life' (MCL). The data, described in Zajdela et al. (2022) ,was made available courtesy of Research Corporation for Science Advancement (RCSA). RCSA organizes conferences in their *scialog* (science dialog) program. It brings together early career researchers as well as experienced scientists. Throughout the conference, attendees interact and are encouraged to work together. At the end of the conference, the researchers submit collaborative research proposals. Each person can be in a maximum of two proposals. Out of up to 30 proposals, 5-10 are selected for funding. We analyse the emerging collaboration network, where each node is an individual and each edge implies that the pair collaborated together. In Appendix B.2, we show results for other three conferences on which we obtained data.

Complexity Global School Interactions. This unweighted, directed network was created using anonymized 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. These messages could be either to students' personal account, or to a faculty member's. If a user talked with another user inside a space (a discussion group or a project group) we also connect them by a link. Finally, we connect two users if they were writing on the same etherpad. This accounts for collaborations between project groups.

The datasets are summarized in Table 1.

4 Results

This section presents the results. We run the model $p_{\text{obs}} = 0.01$, so that in a network of 100 agents, one of them will 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. The number of decision variables

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..

was set at $K = 50$, with a clause density of $\alpha = 2$, meaning that the number of constraints is $M = 100$. These parameters were deemed sufficiently complex to generate a wide distribution of performances across the networks considered.

Furthermore, 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.

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 stabilize.

4.1 Network performance

In Table 2 we list various network parameters for each network we study. Figure 2 compares the evolution of violations and homogeneity across networks. Marked differences emerge.

Table 2: Key network properties and performance metrics.

Network	N	E	D **(%)**	C_D(%)	$f_{\mathbf{inter}}$(%)	ρ **(%)**	CC **(%)**	V	H
Congress Twitter	475	13,289	5.9	21.64	20.8	46.2	22.4	35.4	0.812
Company Emails	167	5,784	20.9	41.77	65.8	87.6	55.9	38.6	0.777
Political Blogs	1,222	19,024	1.3	15.28	7.4	24.3	21.9	44.0	0.699
Conference Collab.	63	224	11.5	9.1	24.1	100	29.5	36.0	0.80
CGS Interactions	64	180	2.8	11.8	14.4	25.6	40.6	48.8	0.646

Note: 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. Violations and homogeneity are calculated after 500 graph updates.

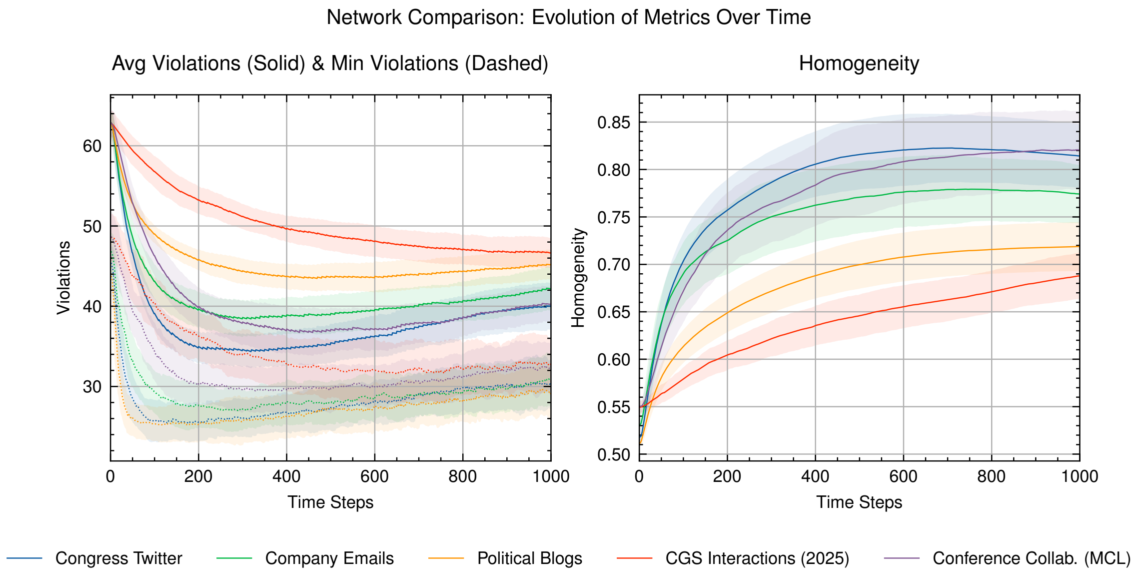


Figure 2: Violations and Homogeneity Over Time

Note: The performance of the model on various empirical networks varies in both average violations and homogeneity. Above, we plot each network's average value 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 stabilize at higher values. The three best performing networks are heterogeneous in terms of size and centrality, but they are all characterized 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, raises the question of why the Political Blogs network, with $N = 1{,}222$ nodes 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 centralized structure which, with a degree centralization 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.

Homogeneity yields a different ranking. 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 ranked last.

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 centralization. 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 centralized. This renders its topology 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 node-level attributes. In particular, we examine the correspondence between a node centrality and its performance in terms of average violations. We focus on the the company email network because, being highly centralized, it exhibits a wide variety of agent centrality (however, the same relationships emerge in the other networks too).

In Figure 3 we plot node centrality and violations, averaged over 50 runs. Centrality, of course, does not change over the run. We notice an exponential decay relationship between a node's centrality and its violations. At low levels of centrality, the negative correlation with average violations is steeply sloped. That is, agents with very low centrality (near zero) have dramatically higher violations. However, the relationship flattens around an eigenvector centrality of 0.025: agents with even modest centrality converge to a stable, low violations level.

The email network dataset includes information on 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). Everyone else works under the CEO at different levels, defined from 1-5. Using node level data and hierarchies, we are able to study how agents who work at different levels in the company perform. This analysis, as well as details of how the levels were calculated, can be found in Appendix B.3. Although the CEO is a node with low violations, the upper management roles do not show correlations with better performance. This could be because of the way the network is structured. The organisation is structured in a way that there are weakly connected nodes even at level 1. Thus, these are workers acting isolated despite proximity to their managers. These causes bottlenecks to the information and diminishes the performance.

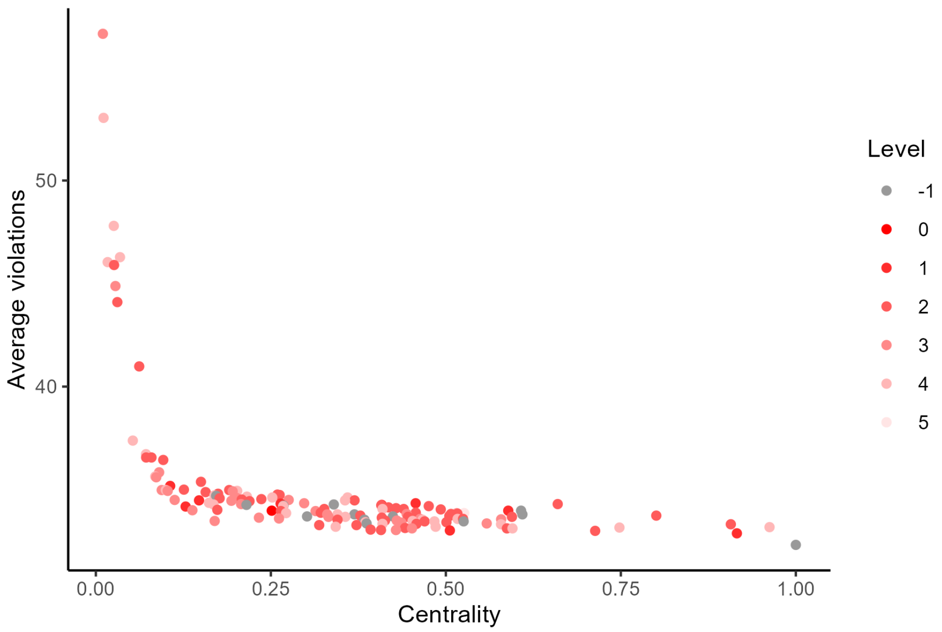


Figure 3

Figure 4: Local analysis

Note: Figure 3 plots average violations for 50 runs of each node within the email network of the manufacturing company against the eigenvector centrality of that node.

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 in the US. This network was introduced in an article aptly entitled 'divided they blog' (Adamic and Glance 2005). We focused on this because it is the most polarized network, exhibiting a peculiar, peanut-shaped topology made of two main communities or clusters: one leaning towards the Democratic Party, and the other the Republicans. Blogs that are politically aligned with one side tend to cite other blogs that are politically similar to them. This creates a peanut-like network, as shown in Figure 5, with considerably more intra- than inter-community links. Such a network presents an excellent case study on how insularity across communities affects collective learning.

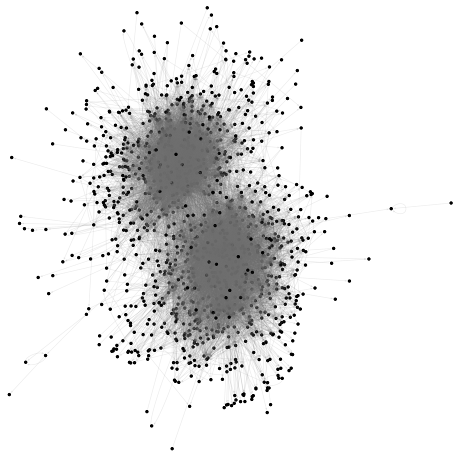


Figure 5: Network visualization of political blogs network, drawn using Graphviz (Gansner and North 2000)

We created individual randomized networks for the political blog networks, yielding the following results.

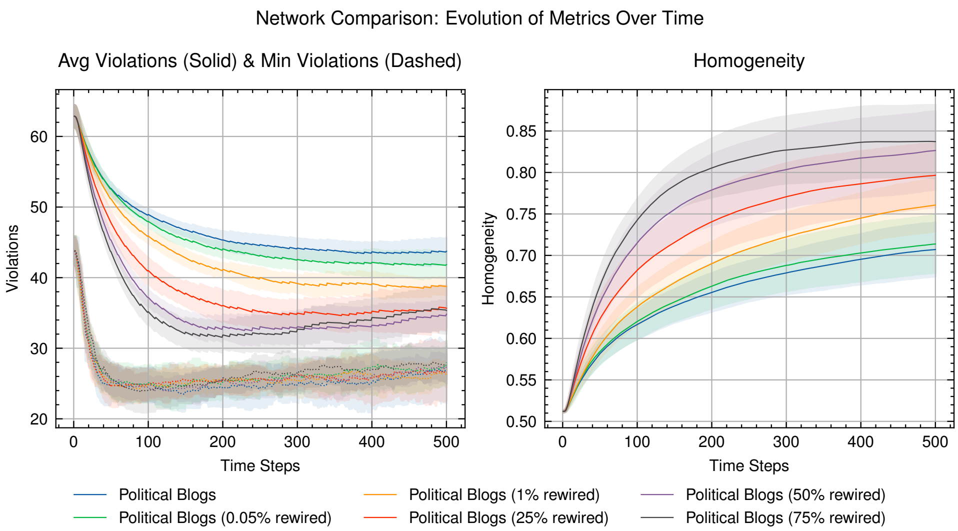


Figure 6: Model results for randomized political blogs network

Note: The figures compare the performance of the baseline political blogs network to versions which were randomly rewired to create more vertical connectivity. Each line shows the network's average across 10 runs.

As we increase the number of inter community links, thereby reducing the network's level of polarization, 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, performance increases up until around 25% rewiring. Beyond this threshold, additional rewiring does not significantly affect performance. This occurs because clauses that are discovered in one cluster can more easily flow to the other. This benefit wanes once a certain threshold of average path length is reached. The result is in line with the classic findings of Watts and Strogatz (1998) and the decreasing marginal reduction in average path length through the cumulative rewiring of a regular ring lattice.

In terms of homogeneity, however, the marginal benefits of rewiring remain meaningful all the way to a full network rewiring. 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 Conclusions and further research

This paper has introduced a simple agent based model of distributed problem solving on a fixed communication network. Agents do not observe a global truth but face a common satisfiability problem: they must choose assignments to a set of binary variables subject to an evolving collection of logical constraints. Knowledge is partial and heterogeneous, and agents update only through local greedy repairs when they learn new clauses, either by direct observation of the environment or from their neighbors. Within this deliberately lean set up, all systematic differences in collective performance arise from features of the underlying network.

Different empirical networks support very different levels of problem solving and consensus even when they are populated by otherwise identical agents. Dense and reciprocal networks, such as the MCL conference, perform best: average violations converge to low levels and homogeneity is high, indicating that most agents identify similar, internally consistent solutions. In contrast, networks that are sparse, fragmented, or sharply divided into loosely connected communities, such as the Complexity Global School interactions and the political blogs, display higher violations and lower homogeneity. In those settings, information on novel constraints tends to remain trapped in local neighborhoods, and agents fail to build a sufficiently rich and shared knowledge base. This result is partly driven by the fact that in our model communication is 'free': for a given network structure, more communication has no disadvantage. Future research may investigate the efficiency of networks in a model with communication costs, where performance might not increase as monotonically with network density.

Second, our node level analysis shows that position in the network matters even when agents are behaviorally identical. More central agents accumulate more information and achieve lower violation counts. Further research is needed to investigate the mapping between such topological structures and organizational roles.

Third, the rewiring experiment on the political blogs network corroborates the result that modest changes in how communities are connected can have large effects on collective performance (Watts and Strogatz 1998). However, while the gains in average violations taper off once paths between communities are sufficiently short, homogeneity steadily increases as the removal of clusters synchronizes the discovery of clause information. This exercise shows that bridging roles and creating ties between communities can enhance group problem solving without requiring any increase in overall connectedness.

These results contribute to our understanding of how the structure of communication networks determines collective intelligence outcomes. An advantage of our approach is that it can be applied to networks from very different domains. The empirical graphs we study are drawn from legislative communication, corporate emails, research conferences and an educational programme. Despite these institutional differences, the same simple model and performance metrics reveal recurring patterns, suggesting that comparatively abstract design principles about connectivity and decomposability can travel across settings.

Future work could introduce richer behavioral heterogeneity, for example allowing agents to differ in their propensity to explore, their memory, or their willingness to revise past decisions. This behavioral heterogeneity may also be explored by using empirical weights which we have ignored in our study. Another avenue for further research would be to endogenize communication, allowing agents to choose whom to consult based on past success, which would link network formation to problem solving.

6 Author contributions

Contributions to this study were collaborative, with the authors providing inputs and engaging in joint discussions throughout. Some stages of the research were, however, primarily carried out by specific authors. F.G. formulated the theoretical model, implemented it on NetLogo and drafted introduction and literature review, methods, results and conclusions. A.Z. led the python implementation, performed the simulations, developed the interactive app, and conducted the empirical network analysis. P.A. contributed to structuring the report and deploying the interactive app. A.D. assisted with some preliminary discussions. AZ, F.G. and P.A. contributed to writing, editing, and approved the final manuscript.

7 Acknowledgments

The authors would like to thank all the faculty at Complexity Global School 2025 held at Bogota, Colombia, organized by the Santa Fe Institute and Universidad de Los Andes. The discussions which took place during the school were significant in development of this project. Brainstorming with Daniel Torres shaped the project in its early stage, while conversations with Joseph Bak-Coleman in Bogota significantly informed our literature survey. This project was formulated in a project group with Daniel Torres. His contribution to the literature survey and outlining the research question is highly appreciated.

The conference dataset was provided by the Research Corporation for Science Advancement(RCSA) via Emma Zajdela. Joan Kim at the Santa Fe Institute assisted in acquiring of the ARCH data used in our analysis.

Finally, the guidance of our mentors, Juan Restrepo, Emma Zajdela and our peer mentor Lavanya Kosuri was invaluable in shaping 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.

Watts, Duncan J and Steven H Strogatz (1998). "Collective dynamics of 'small-world'networks". In: *nature* 393.6684, pp. 440–442.

Zajdela, Emma R. et al. (2022). "Dynamics of social interaction: Modeling the genesis of scientific collaboration". In: *Physical Review Research* 4.4, p. L042001. DOI: 10.1103/PhysRevResearch.4.L042001.

A NetLogo Interface

Figure 7 shows the NetLogo interface used for the interactive implementation of our model. On the left, a set of sliders controls the parameters of the problem and of the environment. The slider `obs-prob` sets the probability that an agent privately observes a new clause in a given tick, while `clause-interval` determines how often a clause in the universal set is replaced. The sliders `K` and `alpha` control the size and density of the constraint

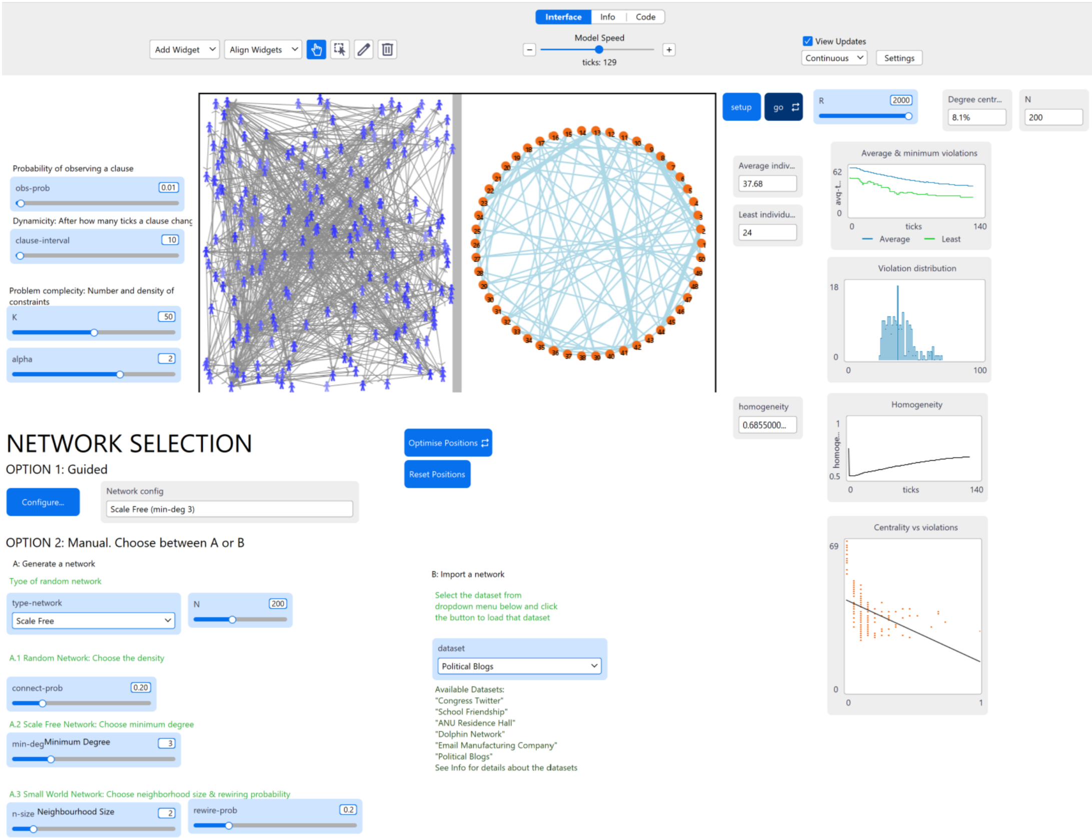


Figure 7: NetLogo Interface

set, respectively. Below, the *Network selection* area allows the user either to generate synthetic networks or to import empirical ones.

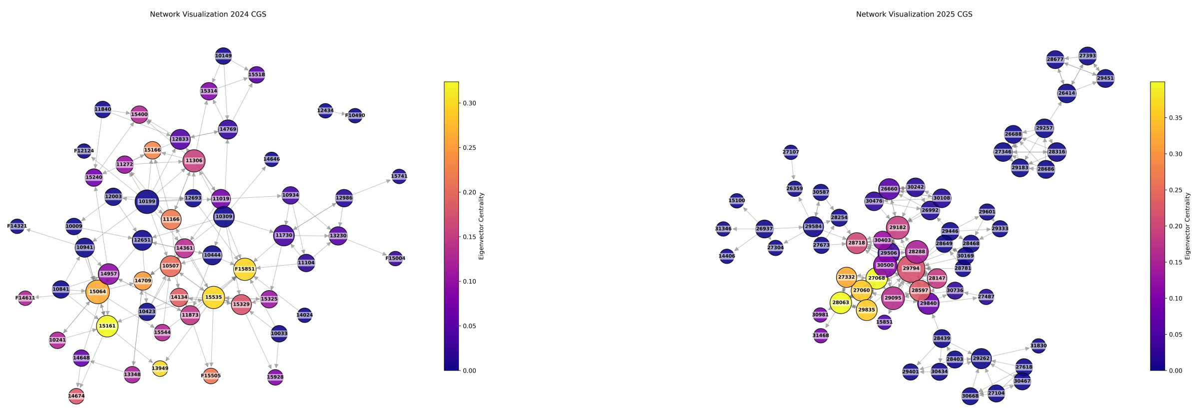
The two large plots at the top show the current layout of the agent network (left quadrant) and a network formed from the decision variables and the clauses that relate them (with one changing every clause interval) (right quadrant).

The right-hand column contains the main simulation controls and output monitors. At the top, a display showing network size `N` and degree centralization. Below, several monitors and plots track performance over time: current average and minimum violations; a time-series plot of average and minimum violations; a histogram of the violation distribution across agents; a scalar monitor for homogeneity together with its time series; and finally a scatter plot of centrality versus violations, updated each tick. Together, these elements allow users to explore how changing the network topology and model parameters affects problem-solving performance and agreement in real time.

B More on empirical networks

B.1 Complexity Global School - a visualisation

Figure 8 shows the network of users interacting through the ARCH platform, provided by the Santa Fe institute. Nodes are sized by their degree and colored by their eigenvector centrality.



(a) CGS 2024

(b) CGS 2025

Figure 8: Complexity Global School Network, based on interactions from the ARCH platform

Figure 9 shows the model's performance for the two interaction networks of CGS editions that we have drawn above. The network for the 2024 school outperforms the 2025 school with better homogeneity among the nodes and fewer violations.

This result is partially because of a disconnected component in the 2025 network, which does not allow the whole network to gather information well. If we compare the network measures between 2024 and 2025 network, we find most measures in a similar range. Both networks have almost save average degree and both networks have a node with maximum degree 10. They have similar number of components and modularity. But the 2024 network has higher inter-community edge fraction (0.29 to 0.13 of 2025). This result again satisfies our result that increasing the inter-community connectivity increases the model's performance.

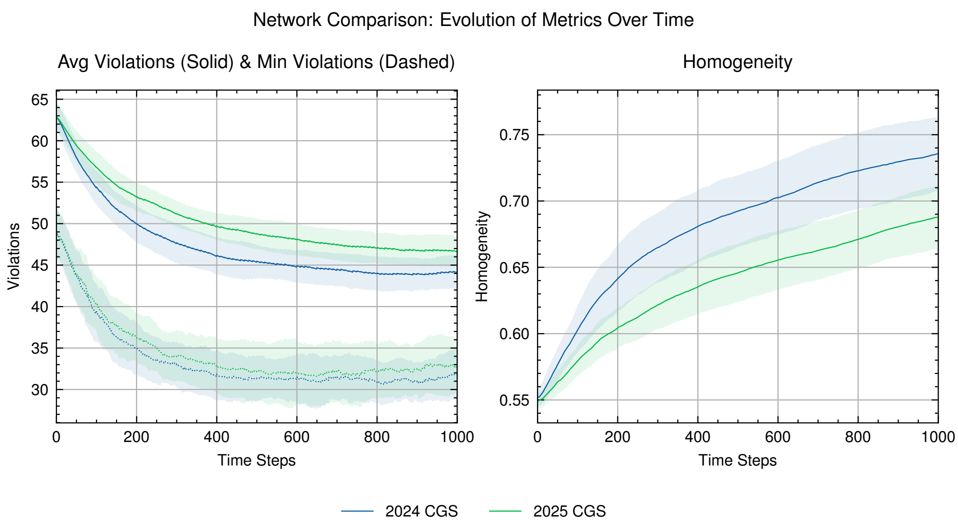


Figure 9: Model's performance for 2024 and 2025 Complexity Global School Interaction Networks

B.2 Conferences - a comparison

We obtained data on four in-person conferences: 2015 Molecules Come to Life (MCL), 2015 Time Domain Astrophysics (TDA), 2017 Advanced Energy Storage (AES), and 2018 Chem. Machin. of the Cell (CMC). These were used to infer collaboration networks, where each node is an individual, and each edge implies that the pair collaborated with one another. See Zajdela et al. (2022) for a richer description.

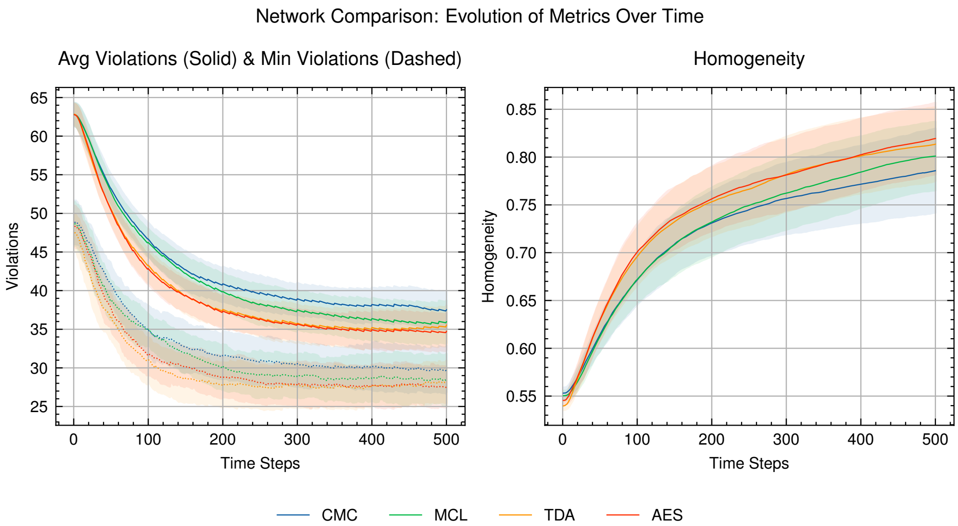


Figure 10: Model's performance on collaboration network from four conferences

B.3 Company emails - a deeper investigation

B.3.1 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.

The dataset includes two pieces of information for every employee: their ID (agent's unique ID) and their manager's ID. The person at the top is mentioned in the metadata as the CEO of the company, which we define as level 0. We calculate each agent'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; 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 11 for a visualization.

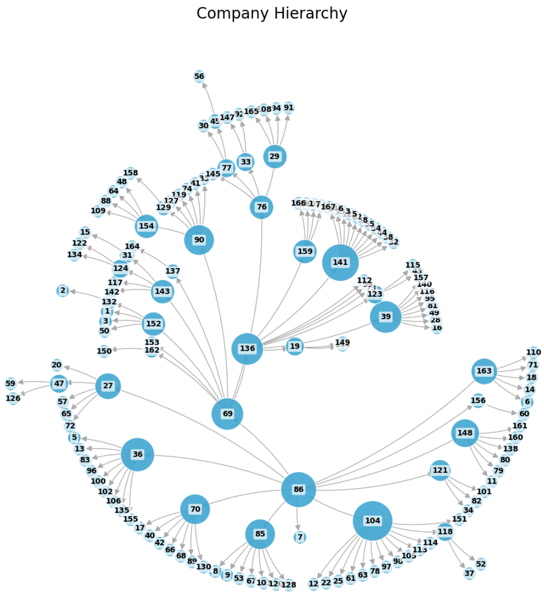
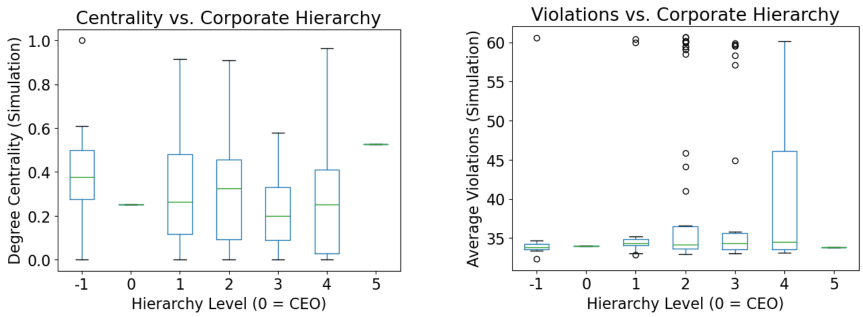


Figure 11: Company Hierarchy

Note: Nodes are sized by their degree.

The dataset includes 13 unconnected nodes, which we label Level -1 to separate them from the main corporate hierarchy. 7 of them are listed as 'technical email account - not used by employees' and 6 of them are 'former employee account'.

B.3.2 Network centrality and performance at different hierarchical levels



(a) (b)

Figure 12: Node centrality and performance in corporate hierarchy

Note: Figure 12a shows node centralities (in-degree) for each hierarchical level of the network, and 12b shows violations for nodes at different levels.

Figures 12a and 12b show 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 violations, i.e. very high efficiency. This is the CEO (shown as Star in Figure 3). Level 1 (upper management) are employees who directly report to the CEO. The box is small, with the median close to the CEO. This points to high efficiency, but there are a few outliers with violation (~ 60). This suggests that some direct reports do not perform as well, despite their high rank.

Middle management roles exhibit greater variation, but interestingly, the best performers even at this level are as good as the CEO. The number of outliers is 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 the 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 has low violation levels, the distribution varies a lot, and average violations are higher.

Tables 3 and 4 provide information on the best and worst performing agents in the network.

Table 3: Best Performing Agents

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 4: Worst 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

B.4 Comparison of different randomization methods

The randomization algorithm discussed in Section 2.2 provides larger performance change, compared to the standard rewiring algorithm. In Figure 13, we plot network performance measures using both rewiring algorithms. In both cases, the same proportion of links were rewired (10%). In the standard algorithm we replace links randomly. In our hierarchical algorithm, we change an intra-community link to inter-community link.

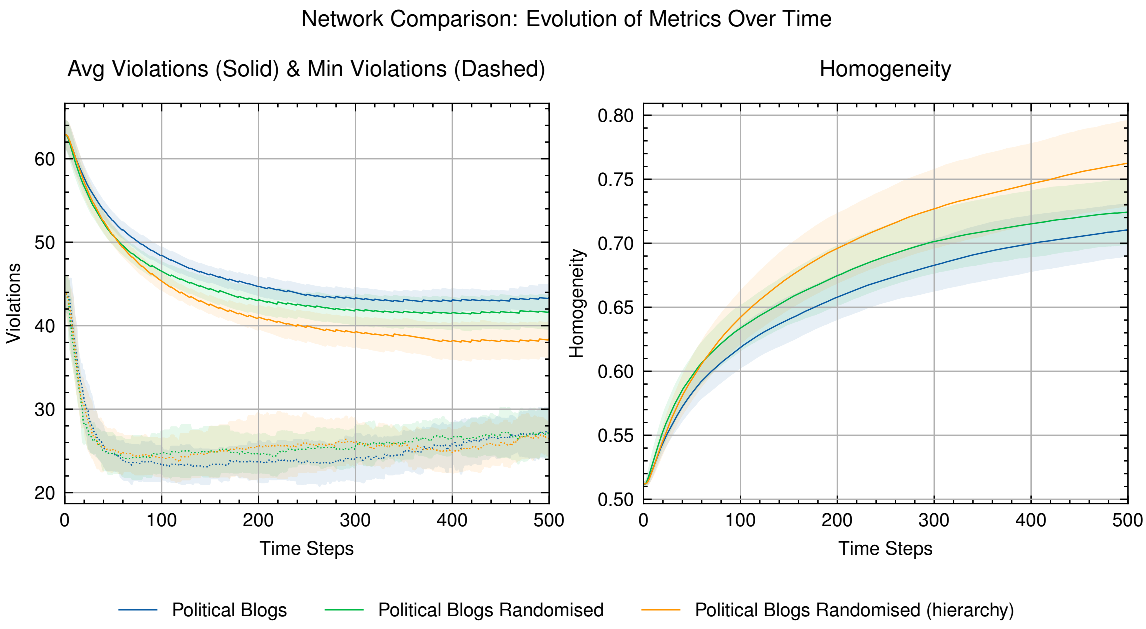


Figure 13: Political Blogs Under Different Rewiring Algorithms

Note: The political blogs network performs as well as the congress Twitter network following randomization. Our algorithm provides better results compared to standard rewiring methods.

When we use 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 average violations seen across the network. The hierarchical rewiring (orange line) improves performance more, achieving the 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.5 Letting constraints evolve every step

In all the empirical analysis in the main text and in the appendix, we let the clauses evolve every 10 steps. This provides a dynamicity to our model but still keeps the network's topology important. If we evolve the clauses or constraints every step, we notice a different trend. Average violations for all the networks converge to a same value within statistical limits.

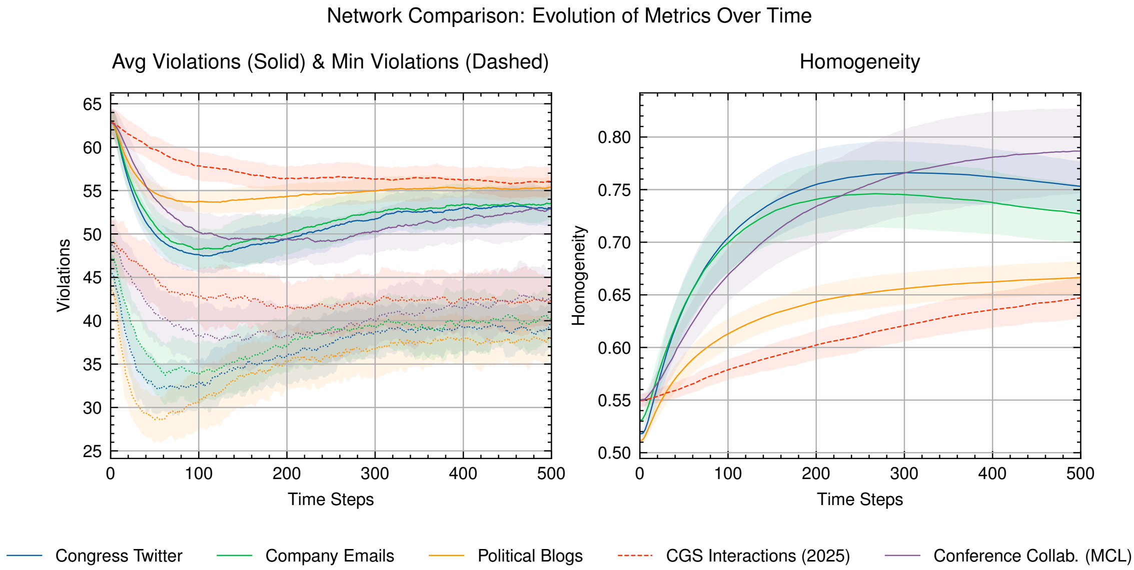


Figure 14: Comparison of the model's result

C Architectures of Problem Solving: the App

We 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 communication graph as an interactive plot. Users can change various parameters of the network and set up 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. The application 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 the front end and NetworkX, Mesa (agent-based-modeling) in the backend. The app is optimized to work best on bigger screen devices, but can be used on smartphones.

Figure 15: QR Code for the website. Scan here to access the web application