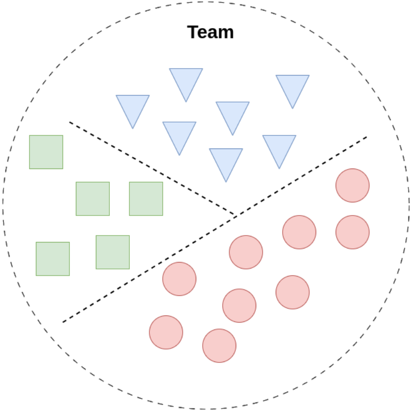
Persuasive Argument Communication in Teams with Strong Faultlines in Maude

Daniel F. Osorio-Valencia
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana Cali
PROMUEVA Project

June 20, 2023

Context

The increasing diversity in organizations and research teams, and the demand for interdisciplinary work, are increasing demographic faultlines within groups.



Context

Effect of demographic faultlines

- Reduces social cohesion and increases relationship conflict, but enhances creativity and innovation [1].
- Strong faultline creates prominent subgroup distinctions, which may give rise to a 'group-split' or polarization [2].

How can we study these effects?

- "To unravel the complex social dynamics shaping consensus, cohesion or disagreement in organizations, researchers have employed the analytical power of computational modelling." [2]

Context

The authors in [2] present 3 computational models that take into account:

- Persuasive-argument communication: argument exchange between individuals influence their opinion.
- Homophily: individuals associate with similar others.

in groups with demographic faultlines. In this work, we present an implementation of the these models using Maude.

Model Description and Initialization

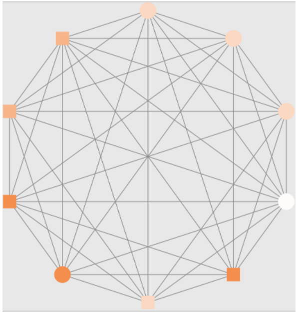


Figure: Taken from NetLogo model in [2]

- Network of agents that communicate with each other.
- Each agent belong to a specific group.
- Interaction between agents change their opinion through time.

Agents Description

The model consists of a Group N of agents, where each agent i:

- An identifier id_i.
- A group identity $g_i \in \{-1, +1\}$, used to represent a strong faultline.
- An opinion in a given time t represented as $o_{i,t} \in [-1, 1]$.
- Depnding on the type of communication, a set of arguments of size S, with $P_{i,t}$ positive arguments and $C_{i,t}$ negative arguments $(S = P_{i,t} + C_{i,t})$.
- Opinion is calculated as:

$$o_{i,t} = 2 \cdot \frac{P_{i,t}}{S} - 1$$

  Positive arguments are in favor of opinion $o_{i,t} = +1$, and negative arguments are in favor of $o_{i,t} = -1$.

Agents in Explicit Argument-Communication

Agent Definition

If argument communication is modelled explicitly the definition for an agent at time t is:

$$ag(id_i, g_i, o_{i,t}, A_{i,t})$$

where:

- Arguments are defined as a tuple (id_a, v_a) where id_a is an identifier and $v_a \in \{-1, +1\}$ defines if the argument is a positive or negative.
- There is a global set of positive arguments Pro and negative arguments Con.
- The set of arguments of the agent at time t is represented as an ordered set $A_{i,t}$ of arguments, where each element must be in $Pro \cup Con$.

For example:

$$ag(4, -1, 0.5, \{(10, 1), (3, -1), (5, 1), (2, 1)\})$$

Agents in Implicit Argument-Communication

Agent Definition

If argument communication is modelled implicitly the definition for an agent at time t is:

$$ag(id_i, g_i, o_{i,t})$$

where the argument set is not used, since arguments are modelled implicitly.

For example:

$$ag(2, 1, -0.75)$$

Agent Initialization

Agent Initialization

For each one of the argument "slots" in the agents argument set:

- Agents with $g_i = 1$ will receive a positive argument with probability w.
- Agents with $g_i = -1$ will receive a negative argument with probability w.

w models how the initial opinion is related to the group identity.

For explicit communication, if $S = 4$, $w = 0.75$ and $g_i = 1$ then the most probable outcome would be:

$$ag(1, 1, 0.5, \{(1, 1), (5, 1), (4, 1), (3, -1)\})$$

For implicit communication, if $S = 4$, $w = 0.75$ and $g_i = -1$ then the most probable outcome would be:

$$ag(1, -1, -0.5) \text{ since } P_{i,0} = 1 \text{ and } C_{i,0} = 3$$

Network Description

- A network is a set of agents $Net = \{ag_1, ag_2, ..., ag_N\}$, where $|Net| = N$.
- The Network is initialized using parameters N and w, to create N different agents with the initialization process explained before.
- $\frac{N}{2}$ agents will belong to each group identity.

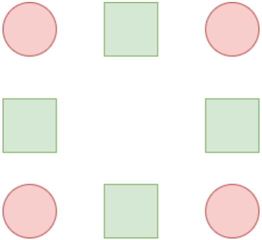


Figure: Initial network with $N = 8$

Model Dynamics

Algorithm General Algorithm

Using parameters N, S and w initialize the network
for $t = 1$ to 10^4 **do**
    for agent $ag_i \in Net$ **do**
        Select partner $ag_j \in Net$ for interaction $(ag_i \neq ag_j)$
        Make interaction between a_i and a_j (a_j influences the opinion of a_i).
        Update a_i's opinion.
    end for
end for

Model Dynamics

Algorithm General Algorithm

Using parameters N, S and w initialize the network
for $t = 1$ to 10^4 **do**
  for agent $ag_i \in Net$ **do**
    Select partner $ag_j \in Net$ for interaction $(ag_i \neq ag_j)$
    Make interaction between a_i and a_j (a_j influences the opinion of a_i).
    Update a_i's opinion.
  end for
end for

Partner Selection: Homophily as Likelihood of Interaction

1. Calculate the similarities of agent ag_i with the rest of agents ag_j:

$$sim_{ij,t} = 1 - \frac{(|g_i - g_j| + h_o \cdot |o_{i,t} - o_{j,t}|)}{2 + 2 \cdot h_o}$$

   where $h_o \in \{0.3, 3\}$ and $sim_{ij,t} \in [0, 1]$

2. Calculate the probability of selecting agent ag_j as a partner:

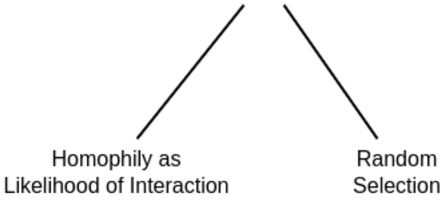
$$P_{ij,t} = \frac{(sim_{ij,t})^{h_s}}{\sum_{j=1}^{N-1} (sim_{ij,t})^{h_s}}$$

   where $h_s \in \{1, 2, 3, 4, 5\}$ and $P_{ij,t} \in [0, 1]$

3. Select probabilistically a partner a_j for interaction.

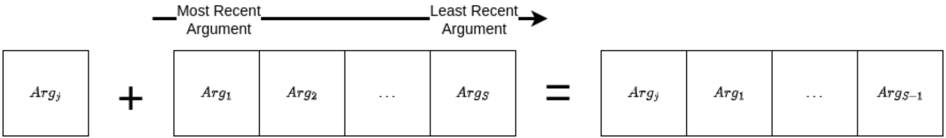
Partner Selection Methods

The other method to select a partner is randomly:

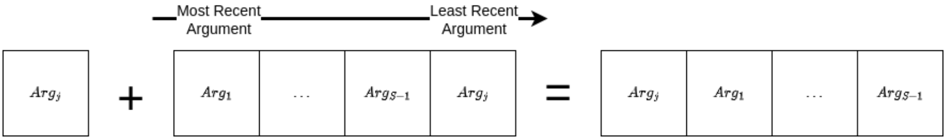


Agent Interaction: Explicit Argument Communication

1. An argument is randomly selected from ag_j's memory and added to ag_i's memory. The least recent argument is then dropped:



2. If the argument is already in ag_i's memory, the argument is placed as the most recent argument, and the rest of the memory is shifted:



Agent Interaction: Implicit Argument Communication

1. Determine the probability that ag_j communicates a positive argument:

$$\text{Probability of } ag_j \text{ communicating a pro argument} = \frac{1}{2} \cdot (o_{j,t} + 1)$$

2. Determine the probability that ag_i drops a positive argument:

$$\text{Probability of } ag_i \text{ dropping a pro argument} = \frac{1}{2} \cdot (o_{i,t} + 1)$$

3. Perform a random experiment that selects the combination of the two events, based on their probabilities.

Agent Interaction: Implicit Argument Communication

4. Compute the magnitude of opinion adjustment $a_{i,t}$:

$$a_{i,t} = \begin{cases} 2/S & \text{if } j \text{ picks a positive and } i \text{ drops a negative argument} \\ -2/S & \text{if } j \text{ picks a negative and } i \text{ drops a positive argument} \\ 0 & \text{if } j \text{ picks and } i \text{ drops the same kind of argument} \end{cases}$$

5. If homophily is not implemented as effectiveness of influence:

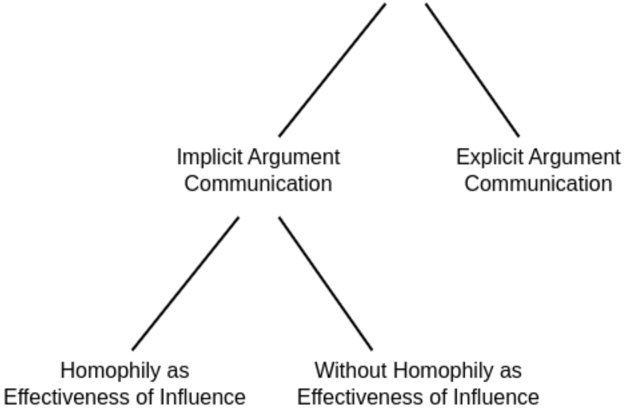
$$o_{i,t+1} = o_{i,t} + a_{i,t}$$

6. If homophily is implemented as effectiveness of influence:

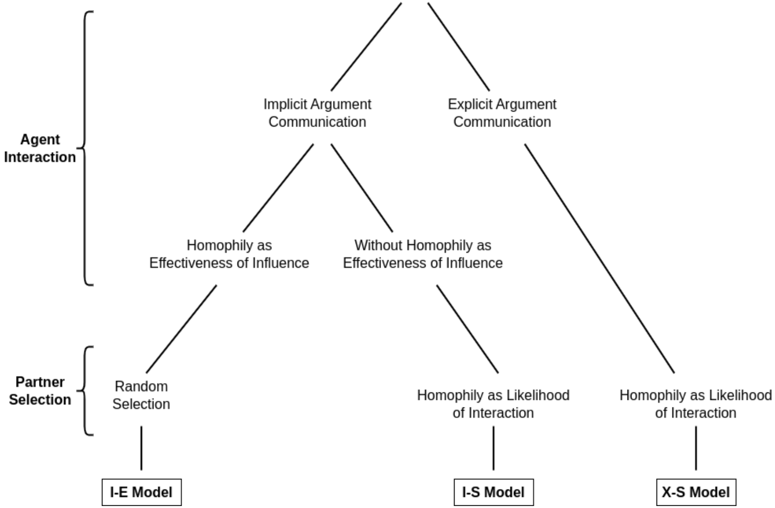
$$o_{i,t+1} = o_{i,t} + a_{i,t} \cdot (sim_{ij,t})^{h_p}$$

where $h_p \in \{1, 2, 3, 4, 5\}$.

Agent Interaction Methods



Computational Models



Property Verification

The properties that want to be evaluated are:

- **Moderate Consensus:** Convergent state where all agents hold the same opinion, but is different from 1 or -1 (only reachable in the X-S Model).
- **Extreme Consensus:** Convergent state where all agents hold the same opinion, and is 1 or -1 (reachable in all models).
- **Between-group bi-polarization:** Corresponds to a number in the interval [0,2], defined as the absolute value of the difference between the average opinions of the two groups.
- **Maximal between-group bi-polarization:** Convergent state where both demographic groups have internally reached extreme consensus on the opposite poles of the opinion spectrum (reachable in all models).

Maude Implementation

Maude is a high performance declarative language, that allows the specification of programs, dynamic systems or logics, and their verification. Using Maude, it is possible to specify, simulate and verify the previously explained models.

Model Specification in Maude

The implementation can be found on `https://github.com/dfosorio/PROMUEVA`. To specify the three models in Maude it was necessary to:

1. Define the required theory (which include functions and data types) used to generate random numbers and probabilistic choices.
2. Define the required theory to define agents, networks and operations over them.
3. Define the 3 dynamic models, using rewriting rules.

Model Specification in Maude

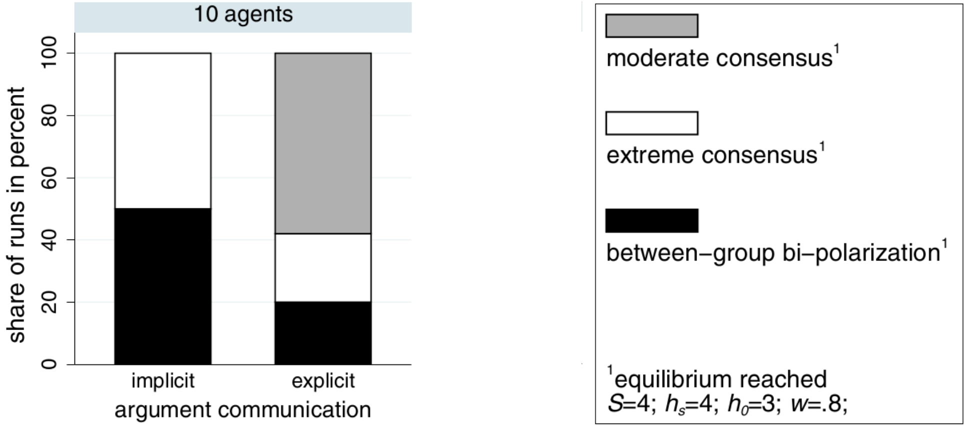
```
--- loop over set of agents randomly (probabilistic since randomPick is used)
rl < {Net}, {Net'}, s(n), ho, hs > =>
   < {Net}, {Net'}, s(n), ho, hs, randomPick(Net') > .

--- probabilistically select a partner for interaction (probability involved in choosePartner)
rl < {Net}, {Net'}, s(n), ho, hs, Ag > =>
   < {Net}, {Net'}, s(n), ho, hs, {Ag}, {choosePartner(Ag, Net, ho, hs)} > .

--- make interaction between agents and update the network (probability in interaction)
--- eliminate the agent1 from the set of available agents
rl < {Net}, {Net'}, s(n), ho, hs, {Ag1}, {Ag2} > =>
   < {replaceAgent(interaction(Ag1,Ag2),Net)}, {Net' \ Ag1}, s(n), ho, hs > .

--- if already looped all agents, reduce the number of iterations, and start over (deterministic)
rl < {Net}, {(empty).Network}, s(n), ho, hs > =>
   < {Net}, {Net}, n, ho, hs > .
```

Model Specification in Maude

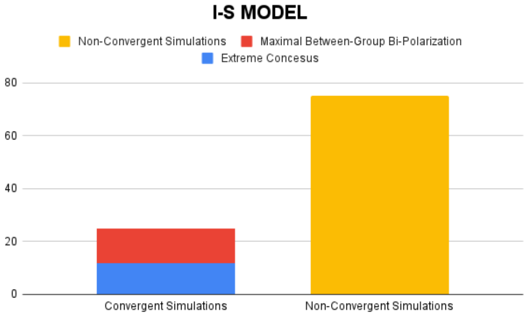
To specify the verification properties and process it was necessary to:

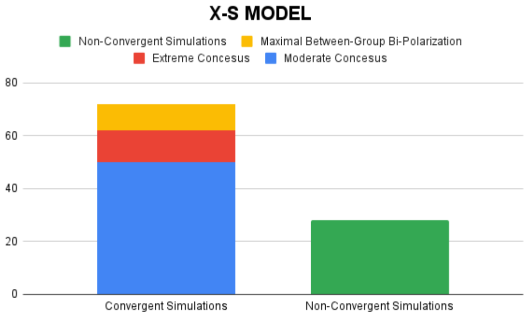
1. Define the required theory to evaluate the final states, and determine if they satisfy one of the previously explained properties.
2. With other programming languages like Python, create a tool that allows to run multiple simulations and recollect the results.

Simulation Results

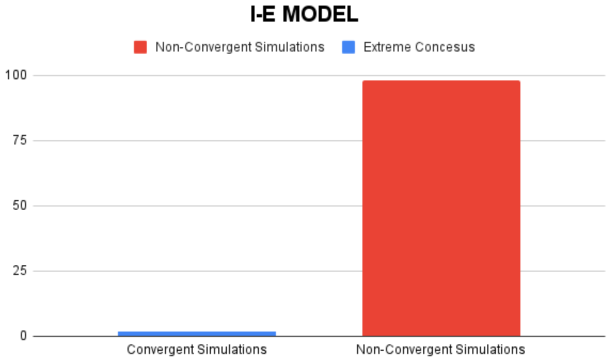


Simulation Results

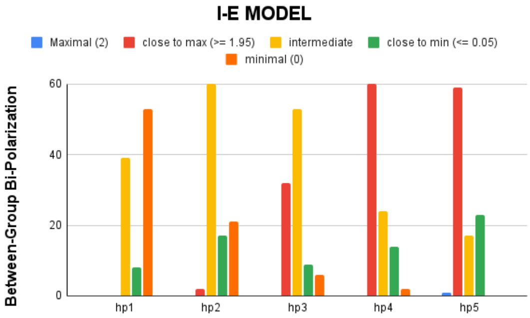


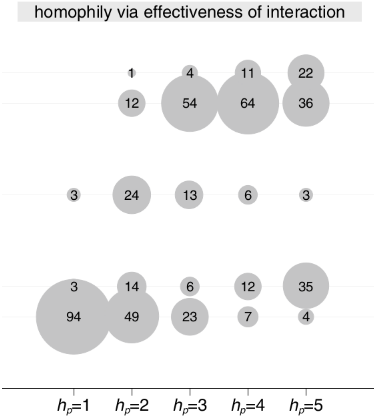


Simulation Results



Simulation Results





Conclusions

- It was possible to implement the three models presented in the article as Maude rewrite theories.
- Even though the models seem to work, the results obtained in the simulations are different to the NetLogo specification results.
- For the Maude specification, 10000 simulation steps was not enough to ensure convergence of the 3 models.
- For experiments with a number $N = 100$ of agents, simulations take a really long time.

Future Work

- Identify why the results of the simulations of the Maude specification, differ from the results from the NetLogo specification.
- Improve the simulation time, in order to execute simulations with 100 agents (probably using parallelism)
- Elaborate a technical report of the implemented Maude specification and the simulation tool.

References I

[1] A. B. Carter and K. W. Phillips, "The double-edged sword of diversity: Toward a dual pathway model," *Social and Personality Psychology Compass*, vol. 11, no. 5, e12313, 2017, e12313 SPCO-0812.R1. DOI: https://doi.org/10.1111/spc3.12313. eprint: https://compass.onlinelibrary.wiley.com/doi/pdf/10.1111/spc3.12313. [Online]. Available: https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/spc3.12313.

[2] T. Feliciani, A. Flache, and M. Mäs, "Persuasion without polarization? Modelling persuasive argument communication in teams with strong faultlines," *Computational and Mathematical Organization Theory*, vol. 27, no. 1, pp. 61–92, Mar. 2021. DOI: 10.1007/s10588-020-09315-. [Online]. Available: https://ideas.repec.org/a/spr/comaot/v27y2021i1d10.1007_s10588-020-09315-8.html.