Decentralized Organizations as Multi-Agent Systems - A Complex Systems Perspective

Technical Report · March 2017

DOI: 10.13140/RG.2.2.17352.42246

1 author:

Shruti Appiah

Decentralized Organizations as Multi-Agent Systems:

A Complex Systems Perspective

Shruti Appiah
University of Waterloo

I take no action yet people transform themselves,
I favor quiescence and the people right themselves,
I take no action and the people enrich themselves,
I am without desires and the people are themselves pristine.

- Lao Tzu, 500 BC [1]

Table of Contents

1. Introduction……………………………………………………………………………………3

2. A case for decentralized and distributed systems…………………………………………5

3. Methods……………………………………………………………………………………………8

3.1 Agent based models

3.1.1 Organizational incentivization mechanisms………………………………8

3.1.2 Agent based models for social conformity…………………………………11

3.2 Graph theoretical models

3.2.1 Analyzing organizations as a social graph…………………………………13

4. Conclusion………………………………………………………………………………………18

5. Limitations and recommendations……………………………………………………………19

6. References………………………………………………………………………………………20

7. Appendix…………………………………………………………………………………………21

1. Introduction

Traditional hierarchical organizations have long been studied, and lay the foundation in many organizational management courses. The advent of a new type of economy - the peer-to-peer economy - calls for a shift in organizational culture to reflect that of a complex distributed system. The primary philosophy behind such systems is to tap into the potential of emergence - “the whole is greater than the sum of its parts” - of self-organized autonomous complex systems.

In the absence of hierarchy in organizations, the social system constituting the members of the organization functions as a complex adaptive system, similar to what is observed on large-scale computer networks such as the internet, or distributed systems like the blockchain. This project frequently compares a decentralized organization to the the blockchain because of its truly distributed nature. On the contrary, the internet does posses same some degree of centralization in the form of centralized clouds, geographically localized service providers, siloed end-user application data, and exclusivity in access to academic work. It appears to be more decentralized than distributed, with a growing trend towards greater centralization with players like Google and Facebook.

Coordination in such organizations may prove challenging since human agents do not behave as simply and rationally as would ants, bees, birds, or servers on the internet. Humans establish social norms in the form of memes, have the tendency to fall prey to cognitive bias, are capable of spreading rumors intentionally or not, and being affected by external political, economic, and environmental factors. These complications give raise to a myriad of faults and nuances which are challenging to account for in a rigid, crisp-logic computer program**.** One would have to be an expert in psychology, sociology, and behavioral economics in order to accurately codify all human tendencies and build a realistic computer simulation.

One way to tackle this problem would be to narrow down the scope of the study sufficiently enough that only a small subset of psychosocial phenomena are exhibited. Researchers have developed some popular social models this way, one of them being the Schelling model of racial segregation.

In this project, an agent-based mathematical model (ABM) to simulate the dynamics of an intra-organizational incentive structure has been developed. Although the model does demonstrate some outcomes of implementing an organizational incentivization mechanism, it does make an assumption that its human agents behave rationally.

For this reason, this report also delves into modelling an organization as a social graph. The reader may borrow some of the presented ideas as a starting point to build their own models, should they wish to study organizational behavior.

2. A case for decentralized and distributed systems

In an organizational context, decentralization could mean anything - from having only team-level management to complete distribution of authority with individual self-managing members. This project examines the latter type of organizational structure. Being inspired by distributed systems like the blockchain, these organizations should similarly offer benefits like intra-organizational transparency and accountability. For an organizational incentivization mechanism, these properties can be achieved by simply implementing the process on the blockchain. This would involve creating a 'coin' and codifying all the rules of the system as smart-contracts on the blockchain. Members of the organization can opt in or out of this system depending on how much they favor the rules of the system. In the future, another process could be deployed to decide the rules in a democratic manner. Currently, there are some blockchain-based systems that offer this functionality. One of them is BoardRoom, which aims to bring the power of governance to everyone. The benefits of distributed system for an organizational incentivization mechanism are discussed below.

Transparency

All the actions and exchanges happening within the system manifest themselves as cryptocurrency transactions on the blockchain. i.e. every action results in a trade of coins between two members. The transactions pass only if they comply with all the rules specified in the system's smart contracts. These can be viewed by any other member on the system free of cost. If implemented in a company-wide private blockchain instance, it would look similar to the transaction history list offered by public services like EtherScan [2], except that this information would only be visible to the members of the organization.

Openness

All the rules of the system are coded in smart contracts and can be stored in a online code repository, thus making it viewable by all members of the organization. It is to be noted that not all members may necessarily be able to interpret the code. This problem can be solved by leaving several human language comments in the code or having a separate constitution that accurately reflects the laws outlined in the smart contract. [3]

Accountability

One of the biggest reasons why scaling organizations turn away from decentralized flat structures is due to the difficulties associated with holding members accountable for their responsibilities. In this system, transaction histories would be stored on the blockchain database forever. This database - being distributed - is viewable by all members of the organization. This extensive timestamped record of everyone's actions can be used as a source of 'truth' to help with conflict resolution.

Fairness is another benefit that implementing this on a codified system brings. Since all members play on an equal ground, members on the system cannot illegitimately favor one member over another. If a bad actor performs any backhanded transactions to another member, they will be caught immediately. Since all the coins are stored on the system, nothing can be hidden from other members. Thus, there is inbuilt fairness and fair competition despite the presence of the inevitable wealth inequality caused by the 80/20 rule and power laws.

Scalability

As opposed to traditional centralized organizations, decentralized systems are not faced with limitation problems that occur with scarce resources, services, and funds. For instance, consider that the company appointed a 'fund allocation manager' to regularly distribute the incentives to members. As the organization scales in size, this person could be limited in terms of their efficiency, the amount of liquid funds available to them, and how quickly they can transmit funds to a member located thousands of kilometers away from them. The fund allocation manager could act as a bottleneck to smooth wealth flows. These problems are avoided in a distributed system because funds are exchanged between individual members (peer-to-peer) and not centrally stored or routed. Since the system is geographically distributed, transactions can happen much more quickly. Transaction speed is often touted to be the biggest value that distributed peer-to-peer economies bring to the table.

Increased reliability

Centralized systems are also more prone to failure due to having a single point of attack. If an attacker targets just the central node or source of control, the resulting failure can trickle down and eventually render the entire system useless. Such centralization failures are akin to the 2008 stock market crash and the October 2016 Distributed Denial of Service (DDoS) attack that brought down Twitter, Github, and several other internet services. [4] Failures such as cascading failures are still possible in a decentralized system since it is a highly interconnected complex network.

Decentralized systems also allow for fault tolerance, especially near self-organized criticality. Due to its inspiration from a blockchain-like decentralized network, the system is subject to failures related to consensus, such as Byzantine faults. Byzantine faults happen when some nodes in the network are traitorous and appear to be in different states (say on/down) to different nodes. This incoherency in information available to different sets of nodes delays the system's arrival at consensus, or can cause dangerous pseudo-consensus, where the nodes in the system appear to be at consensus even when they are not supposed to be i.e, they appear to be at consensus just because some nodes are at a favorable state by satisfying conditions using *false* information provided by the traitor nodes. [5] In a social context, this would be analogous to people exhibiting hypocritical behavior, i.e saying one thing and doing another. A member may be able to participate and claim incentives for a project, yet not have actually contributed to it.

In addition to the above value-adds, distribution of authority is also a strategically advantageous move. In organizations with several layers of management, employees have a tendency to feel apathetic due to feeling like they are being constantly looked over. For centuries since the Industrial Era, economic philosophers have tried to solve the problem of highly specialized jobs causing a feeling of a lack of purpose and yet companies still deploy extensive top-down management structures. Studies have shown that high performers prefer to work in highly autonomous environments. Power decentralization improves employee attitudes, increases autonomy, raises the average individual responsibility and initiative. [6] It is to be noted that flat organizations do not offer any ready benefits past a certain scale (5000 members). Therefore, although the the system itself can be scaled beyond that limit, some key changes need to occur in the structure of the organization which would call for a shift in the topology of the system from a completely flat organization to one with some level of hierarchy.

3. Methods

3.1. Agent based models

3.1.1 Organizational incentivization mechanisms

Consider an organization with several independent projects, each consisting of a team of self-organizing members. Being humans, the members of such an organization act out of self-interest most of the time, and thus devote all their attention to developing their own projects. However, such an organization, like any other, does have organization-wide operational needs such as coming up with a mission statement, organizing company-wide trips and events, creating an intranet, and so on. Certainly, members have a greater interest in working on their own project than to contribute the central projects because there is no incentive for them to do so. How would such an organization garner its members' attention towards these fundamental needs?

In a flat organization without lines of command, it does not make sense to order members to participate in organization-wide projects. Typically, the organization would rely on volunteerism to get such work done. However, traditional volunteerism may be an unreliable means to seek consistent labour, as there would be no accountability, roles and responsibilities would be ambiguous, and the system could be riddled with uncertainties.

Mechanism Design

Mechanism design, or reverse game theory, is used to find the most optimal solution for designing incentives for a group of agents acting simultaneously. [7]

Objective

The objective of this project is to study if and how an organizational incentivization scheme will increase member participation in central organization-wide projects.

Players

Members of the organization.

The game

Members of the organization transact, donate, and accept coins while occasionally post or participate in organization-wide projects.

Initial Conditions

Members start with unequal distribution of coins.

Rules

The rules of this game are outlined in the code *WealthModel.py*, shown in the Appendix (See page 21).

Outcomes

The outcomes of this game are members of the organization participating in projects and being rewarded for completing them.

Model characteristics

Stochastic

In order to reflect the stochasticity of the real world, the model contains several instances where randomness is introduced. Stochasticity is introduced through random selection of agents for receiving donations, random participation in projects, and so on.

Probabilistic

Although this property is similar to stochasticity, it is explicitly stated to illustrate that there are ways in which the probability of outcomes can be pre-specified. This model can accommodate situations where outcomes can be moderately predetermined. For example, if it has been previously determined through data analysis that the organization being examined is highly altruistic, the probability that the *altruism switch* is ON can be increased from the default probability of 0.5.

Default setting with P(agents exercise altruism) = 0.5

```
altruism = random.randint(0,1)
if altruism > 0:
     # perform donations
```

Setting for a highly altruistic society with P(agents exercise altruism) = 0.75

```
altruism = random.randint(0,3)
if altruism > 0:
     # perform donations
```

Probabilities can also be considered in the graph-based study when the relationship between two nodes is ambiguous (See Page 16 of this report). Besides this, the overall transitivity index of the system is also represents a probability, i.e the probability that two nodes in the system are connected.

Markov

The system is modelled as a Markov random field (as in an Ising model of ferromagnetism). Future events only depend on the present state, and not on the chain of preceding past states. Given that the model is stochastic and is run through incremental time steps, the process can be described as being a discrete time Markov chain.

Results

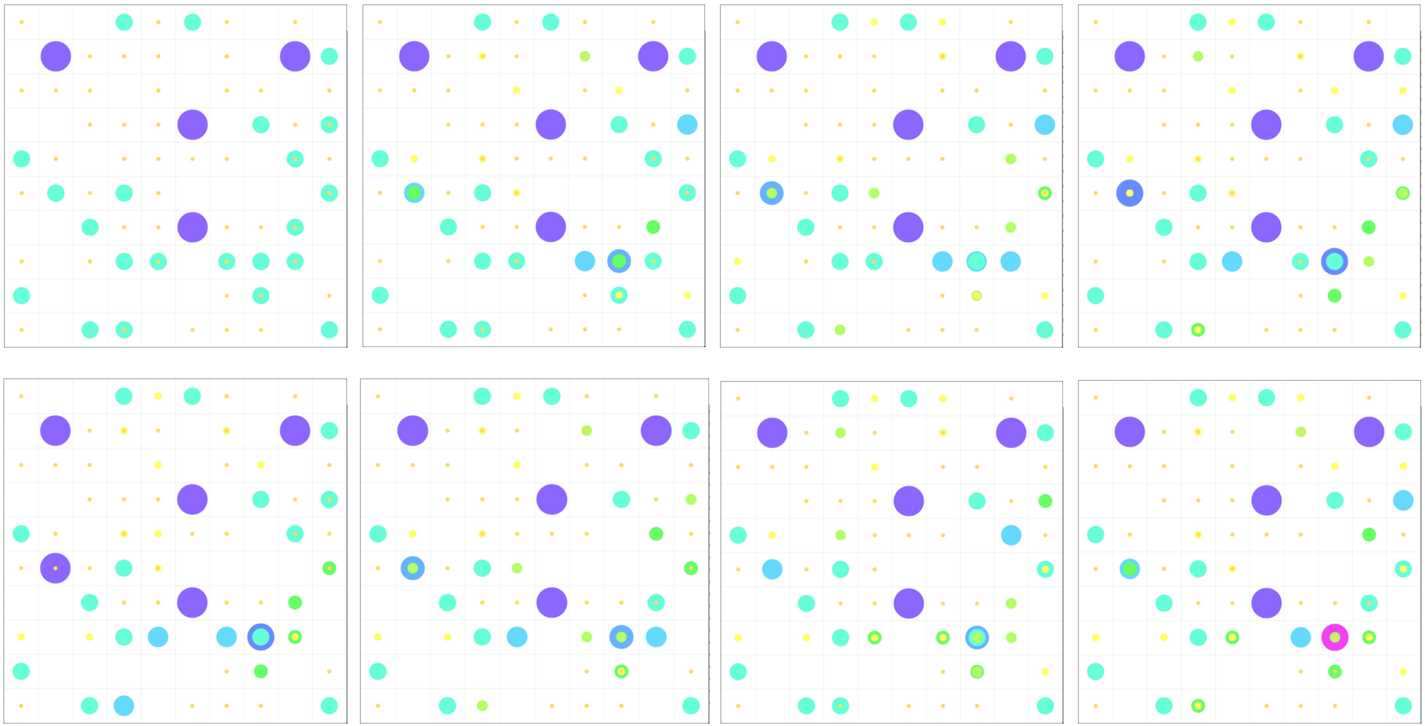

Figure 1: Agent-based model simulating the wealth distribution in an organization with a project incentive mechanism over time. This is a scenario without any taxation schemes. Starting with timestep = 1 with an unequal distribution of wealth, the wealthy agents have the tendency to remain wealthy through time.

Measuring success

Success is measured by comparing the total number of participants, on average, that have contributed to projects. A member can be counted more than once if they have participated more than once. As the taxes imposed on the wealthy and the rewards/incentives for project participation increase, more members contribute to projects overall.

Iteration of running the agent-based model	Total participants at 60 timesteps
Iteration 1	10
Iteration 2	26
Iteration 3	19
Iteration 4	31
Iteration 5	8
Iteration 6	20
Iteration 7	18
Iteration 8	17
Iteration 9	18
Iteration 10	35
Average number of participants at 60 timesteps	**20**

Table 1: Calculating the average net number of participants after a given timestamp

With a tax of 30% of total wealth imposed on any member possessing more than 7 coins, and the reward per project participant being 2 coins, the average number of participants after 60 timesteps is observed to be 20. This is a more desirable result compared to the scenario in which there is no incentive mechanism, hence no taxes and no participation rewards.

3.1.2. Agent based models for social conformity

Humans have the tendency to conform to social norms and accept ideas they hear repeatedly. Certainly, each person's idea acceptance threshold is different and some may be impressionable while others skeptical. Memetics involves the study of propagation of ideas from one human to another in a society in order to form social norms, and eventually, a culture. The word 'meme' is inspired from the word 'gene' which is used to transmit genetic information. [8]

An organization is also constituent of humans who spread and adopt ideas. The agent-based model shown in Figure 2 depicts the idea states of an organization over time. [9] Different colors represent different ideas. The system starts with ample diversity of opinion. With every time step, the agents choose to adopt the majority opinion of their neighbours.

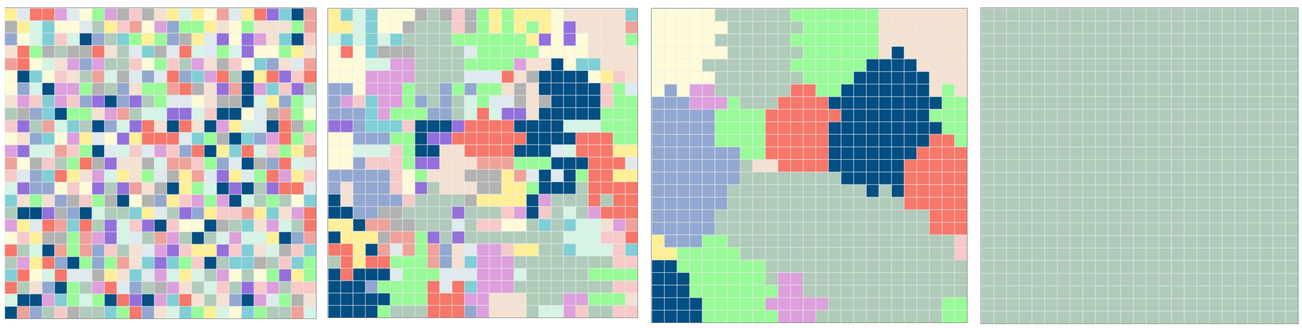

Figure 2 : Agent-based model demonstrating the spread of memes over a population over time. (a) Timestep = 0 (b) Timestep = 3 (c) Timestep = 35 (d) Timestep = 137. Opinion consensus is achieved over a relatively short timespan.

This meme spreading model is inspired from the Schelling model of racial segregation. It appears to be sensitive to initial conditions. In some cases, like Figure 3, the simulation keeps running for several time steps without reaching opinion consensus. This could arise due to there being two or more equally sized majority opinions. When there is such a tie, a person's opinion in the next state is determined by a randomizer that chooses between the two competing opinions of the cell's neighbours. In other conditions, consensus is reached relatively quickly (see Figure 2). However, in all cases, diversity of opinion is lost over time.

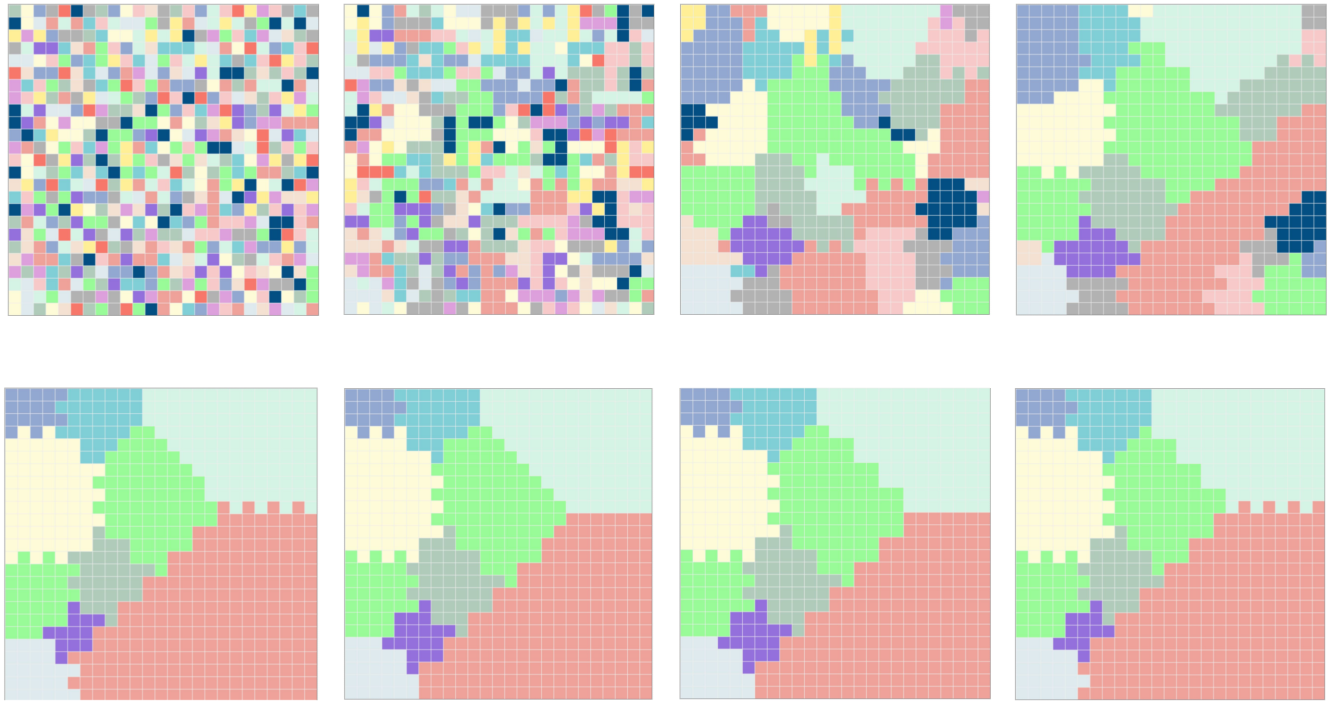

Figure 3: Agent-based model showing spread of memes where opinion consensus is not reached even after 1180 timesteps (bottom right state).

Although it may seem indirect, there is a correlation between widespread opinion and the price of a stock or a cryptocurrency. In March 2017, a cryptocurrency called Ethereum hit its all time high of $53 after a 350% increase since late February. [10] This is partly due to its increased press attention over this time span. Since the value of the organizational incentive coin is also determined by the market - i.e, the members of the organization - it is important to take into account the sociocultural aspect of *reputation* into the model.

Once truthful information transfer is incorporated into the system, it is time to consider corrupt information transfer - such as rumors - as well. Gossip protocols can be studied to formulate the underlying mechanics of rumor spreading. [11] These network protocols have been inspired by the form of communication that happens in closed social systems, i.e. those with a fixed number of members and a boundary, and can be used to reverse engineer complex social interactions by comparing them to simple server-to-server communications.

3.2 Graph theoretical models

3.2.1 Analyzing organizations as a social graph

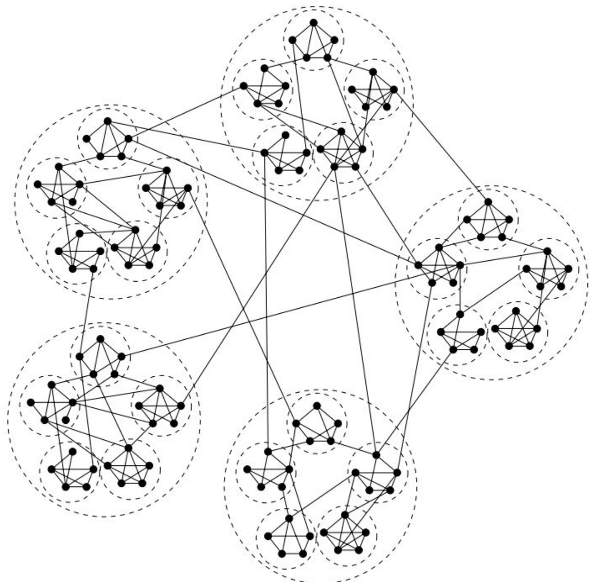

Figure 4: Mesh network topology consisting of several interconnected circles and clusters of sub-circles (Note: number of circle-to-circle connections are arbitrary)

The social networks constituting an organization contain many hubs of highly connected nodes, which may be clustered into groups of highly connected clusters. There is scale free formation of hubs in the network, i.e they can be as big as imaginable, and the network possess behaviour that is independent of scale. These networks also follow power laws and have a non-Gaussian probability distribution of events because conventionally “unlikely” or catastrophic events are more likely than what a linearly-thinking human would expect. Scenarios such as a person holding a disproportionately high amount of wealth, or a node being disproportionately highly connected compared to its peers are not to be seen as anomalies.

Consider a graph with vertices (or nodes) V and edges (or connections) E.

$$G\left(V, E\right)$$

In an organizational model, the vertices are analogous to members of the organizational network, and the edges are analogous to relationships between the members.

$$G\left(M, R\right)$$

If two members are not directly connected through a single edge, their relationship can be defined by the geodesic distance D between them.

$$D = shortest\ path(M_i, M_j)$$

$$\sum_{i=1}^{n-1} f(e_{i,i+1})$$

such that the sum of edge weights over all possible number of vertices n is minimized.
If the graph is unweighted, i.e. all edge weights are equal to 1, then
such that the number of edges between Mi and Mj is minimized.

$$D = shortest\ path(M_i, M_j)$$

Assumptions and simplifications

The following general assumptions about the network have been made:

1. The graph is undirected. This assumption has been made for simplicity and because it more closely reflects the nature of the ideal flat organizational model with no unidirectional lines of command or authority. [12]
2. The edges do not have weights. All relationships are considered to be standard and equal in strength.

Social interactions

Social interactions can be studied by analyzing the flow of information between three nodes or persons in a network. Groups of three nodes are referred to as *triads*. They are the smallest social structures which can represent the nature of the entire organization. [13] It is worthwhile to study these structures while analysing organizational graphs because they provide insight into the behavior and complications that arise upon the addition of an extra node to a dyad. These include the group interaction dynamics, cluster formation, and network faults, all of which can be used to determine the network's global behavior.

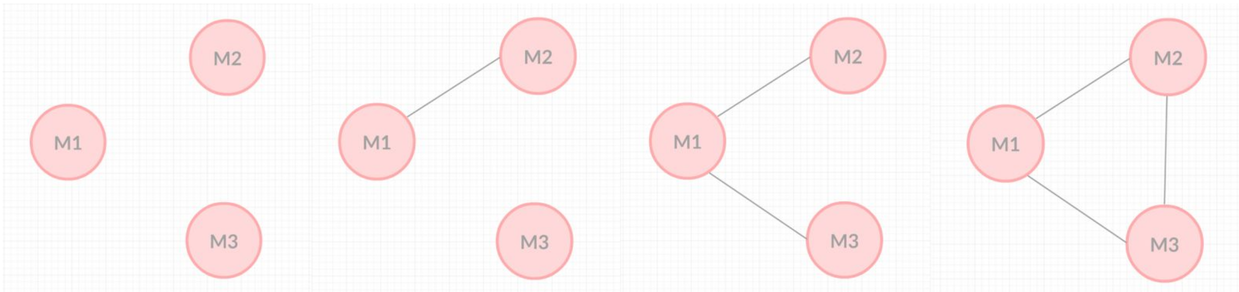

Figure 5: Types of triads. (a) Isolation, nodes don't interact. (b) Couple, two nodes interact and the other is left in isolation. (c) Structured hole, two nodes interact with the same node but not with one another. (d) Cluster, all nodes interact.

Transitivity

Transitivity describes the extent to which the nodes in network tend to form clusters. The organization is assumed to have a transitivity index of 1, and thus is modelled as a complete mesh network. This means "a friend of a friend is my friend", but "a friend of a friend of a friend is not necessarily my friend". Information can transmit from one node to the nodes that are its direct, 1st connections, and also to their connections, i.e its 2nd connections. This forms a 3-edged triad of nodes.

The rationale for this assumption is that the transitivity indices of social networks are distinctively higher than those in a network with connections assigned stochastically. [14]

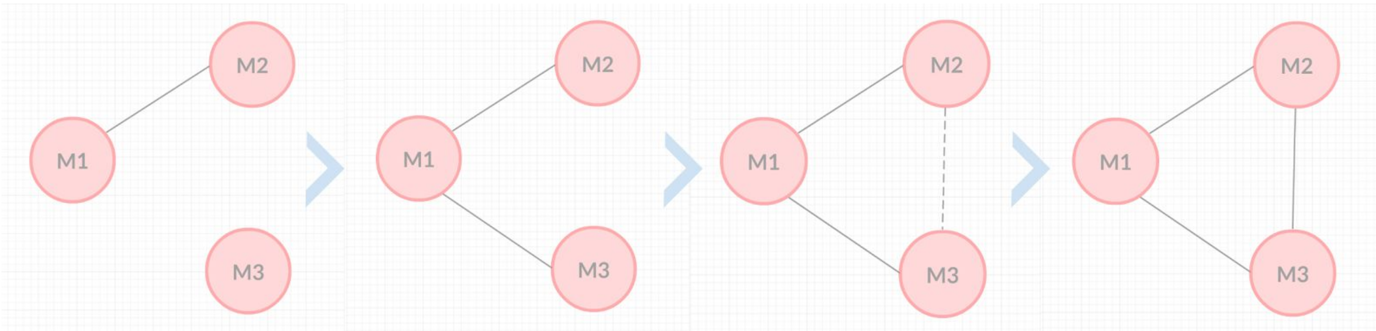

Figure 6: Transition from a coupled interaction to a cluster. (a) Couple (b) Structured hole (c) Partially transitive (d) Cluster

Figure 6 demonstrates information flow as connections between nodes. Starting from the left, M2 transmits information to M1 and M1 shares this information with M3. The graph in (c) demonstrates the situation in which there is some uncertainty in whether the information received by M3 from M1 is the same as the information transmitted from M2 to M1. Assuming that M1 is a good player who does not corrupt the information, the three nodes M1, M2, and M3 all have the same information, thus making it a *closed* triad.

Had the assumption about the transitivity being 1 not been made, the triad is considered to be partially transitive, as shown in (c). There may or may not be a connection between M3 and M2, and the probability of this connection is associated with the transitivity index of the system. If the transitivity index of the system is less than 1, the probability of a triadic closure shown in (d) would be less than 1 as well.

The degree of transitivity of a social network indicates how tightly knit it is. The interconnections between different nodes could be used as a starting point to study the interactions within a system. Based on the frequency and strength of interactions, insights about the relationships that exist within an organization can be derived. This metric is especially useful while devising simulations for the spread of ideas, like the agent-based model described in Page 11-12. The obtained values can be used to determine the probabilities that a person’s opinion would change, since a person is more likely to be influenced by the opinions or ideas of someone that is close to them.

Power dynamics

The organizational model must account for implicit hierarchy by describing roles and minor ego-networks. Ego networks describe the direct connections of a single person or node - called the ‘ego’ - in a network. They consist of the set of directly connected nodes, and the associated edges that connect them to the primary ‘ego’. Even though there are no formally-assigned power holders, informal power structures emerge.

Over time, without deliberate power distribution, these ego networks become stronger and more unbalanced, thus moving the entire organization towards centralization. Power dynamics in a social network can be studied by utilizing directed graphs, particularly by analyzing the types and the frequency of certain types of directed triads.

4. Conclusion

Decentralized organizations with distributed authority offer plenty of benefits and can bring about cohesion through emergence. However, when faced with the challenge of getting members to contribute to certain projects without the use of order or command, the organizational designer is left perplexed. One would have to devise mechanisms to garner the members' attention towards these central projects. This project explored one such mechanism - an incentive scheme. The results of the agent-based simulation showed that the project participation did indeed increase with an increase in the incentives.

In addition to agent-based models, this project also offers a starting point to studying organizational social systems as a social graphs. Triads, which form the atomic units that can describe the behaviour of the entire organization can offer a good starting point to go about designing an organizational model. The census of different types of triads each with a different transitivity occurring in a population can give insight into the frequency and types of interactions in the system.

Before simulating any organizational social system, the researcher has to make a clear distinction between the importance of its central components - the individual persons versus the relationships that exist between people. For a given problem, if studying the individual entities brings more insight, they would pick an agent-based or cellular automaton model. Likewise, if relationships characterize the societal dynamics better, they would choose a graph theory based approach.

5. Limitations and recommendations

Quality of work cannot be measured using the current organizational incentives agent-based model. An empirical study can be performed to compare the quality of completed projects when there are no incentives to the quality of completed projects when there is an incentive scheme. It is expected that the quality declines with an incentive scheme because participants would be motivated by extrinsic motivators, i.e coins, instead of intrinsic, i.e interest and personal satisfaction. However, this hypothesis can only be confirmed through empirical research.

The meme spreading agent-based model makes the assumption that all agents are equally likely to be influenced by the majority opinion. This may not be the case in real life. Several factors determine the probability in which a member would change their opinion, such as their degree of impressionability, the relationship they have with the people holding the majority opinion, and the frequency at which they are exposed to the majority opinion. The latter two factors can be accounted for by using the metrics obtained from the triadic approach.

One of the primary simplifications made in the social graph models is the use of undirected triads. Using directed graphs will provide more insight into the interactions and ego networks occurring in the system. Members can be clustered based on their conditional triad census, which is the number of a certain type of directed triad. Most of the groundwork in social informatics is based on directed triads and the ego networks that are formed as a result.

6. References

[1] "History of Chinese Political Thought, Volume 1", *Google Books*, 2017. [Online]. Available: https://books.google.ca/books/about/History_of_Chinese_Political_Thought_Vol.html?id=Bn19BgAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false. [Accessed: 01- Apr- 2017].

[2] "Ethereum Blockchain Explorer", *Etherscan.io*. [Online]. Available: https://etherscan.io/.

[3] A. Tanenbaum and M. Steen, *Distributed Systems, Principles and Paradigms*, 1st ed. 2017.

[4] N. Woolf, "DDoS attack that disrupted internet was largest of its kind in history, experts say", *the Guardian*, 2017. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.

[5] H. Hsieh and M. Chiang, "Improvement of the Byzantine Agreement Problem under Mobile P2P Network", *IERI Procedia*, vol. 10, pp. 45-50, 2014.

[6] L. Porter and E. Lawler, 111, *University of California, Berkeley* "The effects of "tall" versus "flat" organization structures on managerial job satisfaction".

[7] T. Börgers, *An introduction to the theory of mechanism design*, 1st ed. Available: http://www.econ.yale.edu/~dirkb/teach/521b-08-09/reading/2008-mechanismdesign.pdf

[8] A. Asunción, Memetics: An Evolutionary Theory of Cultural Transmission. Available: http://www.sorites.org/Issue_15/alvarez.htm

[9] D. Masad, Github repository: Project-Mesa, color_patches. Available: https://github.com/projectmesa/mesa/tree/master/examples/color_patches

[10] "Ethereum (ETH) price, market caps etc. | CryptoCurrency Market Capitalizations", *Coinmarketcap.com* [Online]. Available: https://coinmarketcap.com/currencies/ethereum/.

[11] D. Shah, Gossip Algorithms. Available: http://web.mit.edu/devavrat/www/GossipBook.pdf

[12] R.J. Stoll, D. Subramanian, Hubs, Authorities, and Networks: Predicting Conflict Using Events Data. Available: https://www.cs.rice.edu/~devika/conflict/papers/stollsubramanian.pdf

[13] "Introduction to Social Network Methods: Chapter 8: More Properties of Networks and Actors", *Faculty.ucr.edu*, 2017. [Online]. Available: http://faculty.ucr.edu/~hanneman/nettext/C8_Embedding.html.

[14] D. J. Watts and Steven Strogatz (June 1998). "Collective dynamics of 'small-world' networks". Available: http://www.nature.com/nature/journal/v393/n6684/full/393440a0.html

7. Appendix

```
import random
import math
import mesa

from mesa import Agent, Model
from mesa.time import RandomActivation
from mesa.space import MultiGrid
from mesa.datacollection import DataCollector

#Global variables
treasury = 0
economy_scale = 15
project_participation = 0

def compute_gini(model):
    agent_wealths = [agent.wealth for agent in model.schedule.agents]
    x = sorted(agent_wealths)
    N = model.num_agents
    B = sum(xi * (N - i) for i, xi in enumerate(x)) / (N * sum(x))
    return (1 + (1 / N) - 2 * B)


class WealthModel(Model):
    """A model with some number of agents."""
    global treasury
    def __init__(self, N, width, height):
        self.num_agents = N
        self.running = True
        self.grid = MultiGrid(height, width, True)
        self.schedule = RandomActivation(self)
        self.datacollector = DataCollector(
            model_reporters={"Gini": compute_gini},
            agent_reporters={"Wealth": lambda a: a.wealth}
        )
        # Create agents
        for i in range(self.num_agents):
            a = WealthAgent(i, self)
            self.schedule.add(a)
            # Add the agent to a random grid cell
            x = random.randrange(self.grid.width)
            y = random.randrange(self.grid.height)
```

```
            self.grid.place_agent(a, (x, y))

    def step(self):
        self.datacollector.collect(self)
        self.schedule.step()

    def run_model(self, n):
        for i in range(n):
            self.step()
            print("step = ", step())
            """tax_period = step()%10
            if tax_period == 0
                return_tax(self, treasury)"""


class WealthAgent(Agent):
    """ An agent with fixed initial wealth."""

    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)
        self.wealth = 1
        x = random.randint(0, self.model.grid.width-1)
        y = random.randint(0, self.model.grid.height-1)
        print("X Y", x, "and", y)
        if self.model.grid.is_cell_empty([x,y]) == False:
            rich_pos = (x,y)
           rich_receivers = self.model.grid.get_cell_list_contents(rich_pos)
            rich = random.choice(rich_receivers)
            inequality_c =4
            rich.wealth += inequality_c


    def move(self):
        possible_steps = self.model.grid.get_neighborhood(
            self.pos, moore=True, include_center=False
        )
        new_position = random.choice(possible_steps)
        self.model.grid.move_agent(self, new_position)

    ## Daily expenses
    # At every step, the agent makes a transaction with one of their
neighbors
    def daily_transactions(self, coins):
```

```
        cellmates = self.model.grid.get_cell_list_contents([self.pos])
        if len(cellmates) > 1:
            other = random.choice(cellmates)
            other.wealth += coins
            self.wealth -= coins

    ## Altruism
    # At every step, the agent makes a 50/50 choice of whether to donate
money or not
    # If the agent chooses to donate, they donate
    def donate_money(self):
        neighbours = self.model.grid.get_neighborhood(self.pos, moore=True,
include_center=False
        if len(neighbours) > 1:
            altruism_c = 3
            if self.wealth > altruism_c:
                altruism = random.randint(0,1)
                if altruism != 0:
                     for i in neighbours:
                        poor_cell_choice = random.choice(neighbours)
                        poor_cell_contents =
self.model.grid.get_cell_list_contents([poor_cell_choice])
                        if len(poor_cell_contents) != 0:
                                                        poor =
random.choice(poor_cell_contents)

         # If my neighbour's wealth is less than 20% of my wealth,
         # I will donate to them an arbitrary sum of money less than 30% of
my wealth
                            if poor.wealth < 0.2*self.wealth:
                                print("Oh no, my neighbour is poor!", poor)
                                max_donation = int(round(0.3*self.wealth))
                                donation = random.randint(0, max_donation)
                                poor.wealth += donation
                                self.wealth -= donation
                                break

                else:
                    pass

    ## Taxes
    # At every step, the agent's wealth is checked.
```

```
    # If their wealth exceeds a certain amount, they are taxed 30% of their
wealth.
    # Upon paying tax, the agent is admitted to the Committee and can post a
public project
    def collect_tax(self):
        global treasury
        tax_c = 7
        if self.wealth > tax_c:
            tax = math.floor(0.3*self.wealth)
            treasury += tax
            self.wealth -= tax
            print("TREASURY NOW =", treasury)


    ## Reward agents
    # Agents who have participated in projects are rewarded with a bounty for
completing the project
    # Bounty coins are taken from the Treasury
    def project_reward(self, width, height):
        global treasury
        global project_participation
        treasury_c = 6
        if treasury > treasury_c:
            self.grid = MultiGrid(height, width, True)
            x = random.randint(0, self.grid.width-1)
            y = random.randint(0, self.grid.height-1)

            if self.model.grid.is_cell_empty([x,y]) == False:
                position = (x,y)
                potential_receivers =
self.model.grid.get_cell_list_contents(position)
                receiver = random.choice(potential_receivers)
                reward_c = 2
                receiver.wealth += reward_c
                treasury -= reward_c
                project_participation += 1
                print("TOTAL PARTICIPANTS = ", project_participation)

    def step(self):
        #self.move()
        if self.wealth > 0:
            expenditure_c = 1
            self.daily_transactions(expenditure_c)
```

```
            self.donate_money()
            self.collect_tax()
            self.project_reward(economy_scale,economy_scale)
```

All source code and running instructions can be found at
https://github.com/ShrutiAppiah/Simulating-an-Economy-ABM