University of Zurich

Agent-based modeling for Business, Economics, Informatics and Social Science

Final Project

Evolving Communication in Agent-based Models

Instructor: Prof. Dr. Claudio J. Tessone
Semester: FS 2021
Name: Zermin, Adrian
Matr.-Nr.: 20-731-956
E-Mail: adrian.zermin@uzh.ch

Date of Submission: 28th June 2021

Contents

1 Introduction **1**

2 Background **2**

2.1 Simultaneous Move Games 2

2.2 Finite Automata 3

2.3 Genetic Algorithms 4

3 The Model **5**

3.1 Agent Representation 6

3.2 Agent Adaptation 7

3.3 Necessary Adjustments to the Games Played 8

4 Results **9**

4.1 Evolving Communication in the Prisoners' Dilemma 11

4.2 Impact of Exogenous Variables 13

4.3 Evolving Communication in the Stag Hunt 13

5 Outlook **17**

1 Introduction

Communication denotes the act of developing meaning among entities or groups through the use of mutually understood signs, symbols or semiotic conventions [Wikipedia contributors, 2004]. Communication can be found throughout a wide array of systems, from biological, social or economic. It is used to promote organization and performance in agent groups of all kinds, be the intricate mating dances developed by the tropical birds on Galapagos or the small interactions we have in our every day lives, with our colleagues, friends or family. Communication and its patterns and rules can also come in various different forms, they can be deliberately and carefully created in the form of a computer communication protocol like HTTP or develop more organically through learning and experience like the almost ritual-like behaviour that can be found in bars and clubs all over the world, driven by the hopes of finding a mate.

This paper and the model presented herein are going to study the endogenous emergence of communication by developing a model of adaptive agents that play a set of single-shot simultaneous move games. Simultaneous move games have proven themselves a useful tool for modeling social dynamics and have been widely studied regarding optimality and strategic behaviour. Adaptation will be the key driving factor behind the emergence of communication in this agent-based model. In order to ensure the endogeneity of the emerging communication, agents will simply be given access to a standard set of symbols, leaving it up their own devices to assign meaning to them and the only impact communication has is the reaction it induces in the opponent within the context of the simultaneous move game. Also, since we are interested in studying the communication that emerges within a group of agents, not between specific agents, agents shall not be able to identify their opponents by any other means than the exchanged communication patterns.[J. H. Miller et al., 1998]

		P2	
		Cooperate	Defect
P1	Cooperate	3, 3	0, 5
	Defect	5, 0	1, 1

		P2	
		Stag	Hare
P1	Stag	20, 20	10, 18.5
	Hare	12, 12	10, 18.5

Figure 1: The Prisoners' Dilemma and Stag Hunts' Payoff Matrices

2 Background

In order to provide a thorough foundation of the concepts and principles used, the following sections are going to provide a general view of the four main concepts used to model endogenously evolving communication in agent-based systems. While most of these concepts have been thoroughly investigated by the scientific community, the methods and underlying concepts employed vary depending on the researcher, thus making it necessary to establish a baseline for the theoretical concepts used in the development of any agent-based model.

2.1 Simultaneous Move Games

A *simultaneous move game* is, as its name implies, a game of any kind where players choose their moves at the same time. Maybe the most commonly known example here would be rock-paper-scissors and its variations. Much like rock-paper scissors can be found, with some variation, across many different cultures, the study of simultaneous move games has applications across many different fields, such as biology, psychology or economics. For this model we will be focusing on 2x2 one-shot (non-repeated) games, these are single instances of 2 player games where both players simultaneously choose one of two moves. This class of game can be modeled through a *payoff matrix*, wherein the reward each player receives is shown depending on his opponents' move.

In Figure 1 such payoff matrices are shown for the two games we will be focusing on in this paper: Prisoners' Dilemma and Stag Hunt. [1] The specific payoff values for the Prisoners' Dilemma are modeled after [J. H. Miller, 1996] and after [J. Miller and Moser, 2004] for the Stag Hunt.

[1] Payoffs are ordered P1, P2

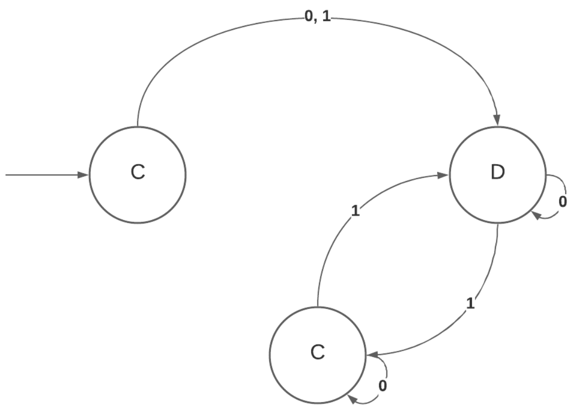

Figure 2: A simple Moore Machine

The Prisoners' Dilemma core feature is that for each player, the reward received is bigger if he plays 'Defect', regardless of what the other player picks, thus making $(D(efect), D(efect))$ [2] the most common outcome. The dilemma here being that the average payoff for both agents would be higher if they both played 'Cooperate'.

As for the Stag Hunt game, it has its origins in [Rousseau and Cranston, 1984], in which it was postulated as a common interest game,where players should aim to coordinate with the other player in order to achieve the outcome (H, H), which is more rewarding than any other outcome for both of them.

2.2 Finite Automata

Finite automata are a class of mathematical models used to describe simple computer program behaviour, based on specific inputs and outputs [J. H. Miller, 1996]. For the model described, we will more specifically be looking at textitMoore Machines as described in [Moore., 1958]. In order to more easily introduce the concept of Moore Machines, we will first look at a very intuitive representation of Moore Machines via their textittransition diagram, as seen in figure 2. Every state, represented by circles, is assigned an action, in this case from the set $\{C, D\}$. These actions are the output our machine provides. Depending on the input, here shown as binary integers, the state of our machine can change and with it, the action it provides. The first state is designated as the starting state with an arrow.

[2] Game outcomes are shown as (Move Player 1, Move Player 2)

	0	1
$\rightarrow 0$	1	1
1	1	2
2	2	1

Figure 3: The sample automatas' transition table

Thus, when our machine receives a the sequence $[0, 0, 0, 1]$ as input, we will receive C as an output.

A more technical representation of such a Moore Machine is provided by the four-tuple $(Q, q_0, \lambda, \delta)$, where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\lambda : Q \rightarrow S_i \in \{C, D\}$ is the function that maps the internal state of the machine to the output and $\delta : Q \times T_i \rightarrow Q$ is the function that maps the current internal state to the next, based on the provided input ($T_i \in \{0, 1\}$). [J. H. Miller, 1996]

Another, equally valid, representation of Moore Machines is given by their so-called *Transition Table*. To generate a transition table, each state is assigned a row, each possible input is assigned a column. The content of the respective cell gives the next state, based on the current state and the received input. The transition table for our sample automata is shown in 3. The states are indexed and the initial state is marked accordingly. In order to represent the outputs, a simple *output list* can be used as reference for the indexed states, ($[C, D, C]$ in our example).

2.3 Genetic Algorithms

Genetic Algorithms are a class of population-based search algorithms, whose basic ideas stem from Charles Darwins' theory of evolution [Darwin, 1859]. They were developed by Holland [Goldberg and Holland, 1988] and while they encompass many variations, these variations share several common attributes [J. H. Miller, 1996]:

- A large pool of possible solutions, S
- A *population* $P \subset S$ represented as well-defined structures $p_i \in P, i \in \mathbb{N}$ with $\mid P \mid = n$

- An environment which allows us to receive information on each structures performance i.e. its *fitness*

- A procedure to generate new populations

- A modification strategy for the structures based on genetic operators i.e. a *mutation* method

Once these prerequisites are given, we can postulate the following procedure: First, a population is randomly selected. Second, every structure is tested against every other structure and its fitness is evaluated. A certain number of top performers are selected from the population and placed in a new population. Next, two structures are chosen from the population to reproduce, the fitter of which is then chosen to be altered by the mutation method and placed in the new population. This is repeated until the size of the population is n once again. The new population then undergoes the process from the second step on, until a certain number of iterations is reached. [J. H. Miller et al., 1998] The developed solution for the population is going to promote better performing solutions and genetic algorithms in general have been shown to provide effective solutions for highly nonlinear systems [Frantz, 1972], making the procedure extremely well suited for developing population-based adaptive models.

3 The Model

Now that we have established an understanding of the concepts used to develop our model, we will be defining its properties. The approach highlighted here is modeled after J. H. Miller et al., 1998. The aim of our model is to show and analyse endogenously emerging communication between two agents in simultaneous move one-shot games. In order to do this, we are going to define a set of agents $P = \{1, ..., n\}$, n being exogenously given, whose representation is highlighted in the following section.

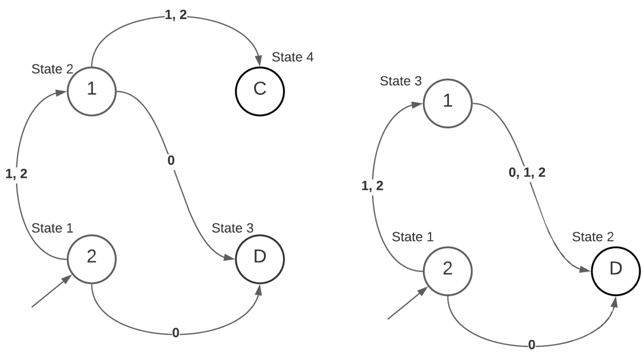

Figure 4: Left: Agent A Right: Agent B

3.1 Agent Representation

Each agent is represented by a Moore Machine that defines its' strategy in a simultaneous move game. The number of states assigned to each agents' Moore Machine, c, can be viewed as defining their computational capacity, since more complex Moore machines allow for more complex strategic behaviour [J. Miller and Page, 2007] and changes the model dynamics fundamentally, making the maximum capacity of every agent another input variable of our model. Agents are going to be playing either the Prisoners' Dilemma or the Stag Hunt. Before playing, agents are given access to a set of *communication tokens*, $T = \{0, 1, ..., t\}$ which they can simultaneously exchange with their opponent at every time step, allowing for communication to occur before each agents chooses their final move. In the suggested approach, the '0' token has a special meaning, it only gets sent out once an agent has chosen a move from the set of final moves provided by the game played, making it a signal that an agent has 'locked in' their final move. Once an agent has chosen their final move, it will continue to send out the 0 token until its opponent chooses their final move. Again, the number of accessible tokens is given when defining our model, with the communication token 0 always being present. In our example games, the sets of final moves are $\{C, D\}$ for the Prisoners' Dilemma and $\{S, H\}$ for the Stag Hunt. These communication tokens are then given to our agents (represented as Moore Machines) as input.

Since the dynamics of an individual game are rather complex to understand at first glance, let us look at the interaction between two sample agents and their automata representations shown in Figure 4, before playing a game of

Prisoners Dilemma. In the example, the agents are able to communicate with tokens $t \in \{0, 1, 2\}$ Initially, Agent A and Agent B exchange tokens 2, which puts Agent A in State 1, Agent B in State 3. Next, both Agents send out token 1 , which puts Agent A in state 4, which nets him the final move 'C(oordinate)', so he sends out the token 0. As for Agent B, he also selects his final move, 'D(efect)', send out token 0 and the game ends, with the outcome (C, D), which nets Agent A a payoff of 0, and Agent B a payoff of 5.

3.2 Agent Adaptation

Now that we have defined our agents and their interactions at a micro-level, we can define our models' behaviour and the agents' adaptation at a population level through a genetic algorithm.

In this case, the pool of possible solutions is the space of possible tactics for the predefined game that is being played. We can then create a random population P of n Moore Machines with up to c states, represented by their transition matrices, which map any possible state and token received to a new state and their action mapping arrays, which contain an action, that can either be to lock in a final move and continuously send the 0 token until the game ends, or send a communication token from $T = \{0, 1, ..., t\}$. For every agent, the transition matrix and action mapping array are created the following way: For every state, the matrix is given a row, for every possible communication token, it is given a column. each value in the matrix is then randomly selected from the pool of possible states (including the present one). As for the action mapping array, for every possible state we either (with a 50% probability each) assign a final action or randomly select a token from $T \backslash \{0\} = \{1, ..., t\}$. An initial state is then assigned with with a uniform probability across the possible states.

In order for us to evaluate each agents' fitness, we hold a round-robin tournament where every agent plays every other agent, in the predefined game, accumulating their payoff. Next, in order to generate our new population, we randomly select two agents from our current population, compare their average payoff and place the better performing agent in the new population. Once an agent is selected for the new population, it is transformed via the mutation procedure with a 50%

		P2		
		Cooperate	Defect	Undecided
P1	Cooperate	3, 3	0, 5	2, -5
	Defect	5, 0	1, 1	2, -5
	Undecided	-5, 2	2, -5	-5, -5

		P2		
		Stag	Hare	Undecided
P1	Stag	20, 20	10, 18.5	12, 0
	Hare	12, 12	10, 18.5	12, 0
	Undecided	0, 12	0, 12	0, 0

Figure 5: The Prisoners' Dilemma and Stag Hunts' adjusted Payoff Matrices

probability. The mutation procedure is implemented as follows: First, a random state is selected, then, again with 50% probability each, either a random transition from that state is changed or the action associated with the state is changed. Which state the new transition goes to or which action is newly assigned is selected with the same approach as described for the random strategy generation procedure highlighted above. Each instance of this selection and mutation process is carried out n times, ensuring that the population remains the same size throughout the models' run time. One iteration of this process is also called a *generation*. After each generation, agent payoffs are reset and a new generation begins.

3.3 Necessary Adjustments to the Games Played

The introduction of communication into the realm of simultaneous move games makes an adjustment to their structure necessary. Namely, we need to restrict the number of tokens agents are allowed to exchange before both players have selected a final move, otherwise, due to the randomness used to generate our population and to mutate our agents, we would have some games that go on forever, because one or more of the agents' strategies never assigns a final move. [J. H. Miller et al., 1998] The adjusted payoff matrices for both games and the new 'Undecided' outcome as implemented in this model are shown in Figure 5.

The model description above leaves us with six exogenous variables that impact what kind of communication happens between our agents: $n \in \mathbb{N}$ is our population size, $g \in \mathbb{N}$ denotes the number of generations we run our model for, $G \in \{'SH', 'PD'\}$ denotes the kind of game that will be played (Stag Hunt or Prisoners Dilemma), $c \in \mathbb{N}$ the computational capability of each agent, $t \in \mathbb{N}$, the number of tokens that are available to our agents to communicate and finally the $timeout \in \mathbb{N}$ which sets the limit of maximum tokens exchanged in a single game.

4 Results

The following section is going to highlight the results achieved from running the previously specified model. All computations took place on an r5.large instance hosted by Amazon Web Services [3] rented through Jetbrains' Datalore online computing service for Jupyter notebooks. For each of the two games we focused on, we will first be looking at their communication dynamics for a single model instance with $n = 50$ agents, $g = 1000$ generations, $c = 4$ as the maximum states available to any one agent for computation, $t = 3$ tokens by which the agents can communicate and $timeout = 100$. However, since our analysis is focused on the communication occurring before games with *regular outcomes*, i.e. the games where both players reach a final move, and because the non-regular games have a much higher number of tokens exchanged, thus skewing our data, we will be excluding them from our analysis for the most part.

Afterwards we will be looking at the impact of modifying the exogenous variables for each of these games, with a focus on the computational capacity of the agents and the number of tokens given for communication.

Lastly, we will be comparing the results from the two games as well as make inferences about how communication can help in both games to achieve a higher average payoff for every agent.

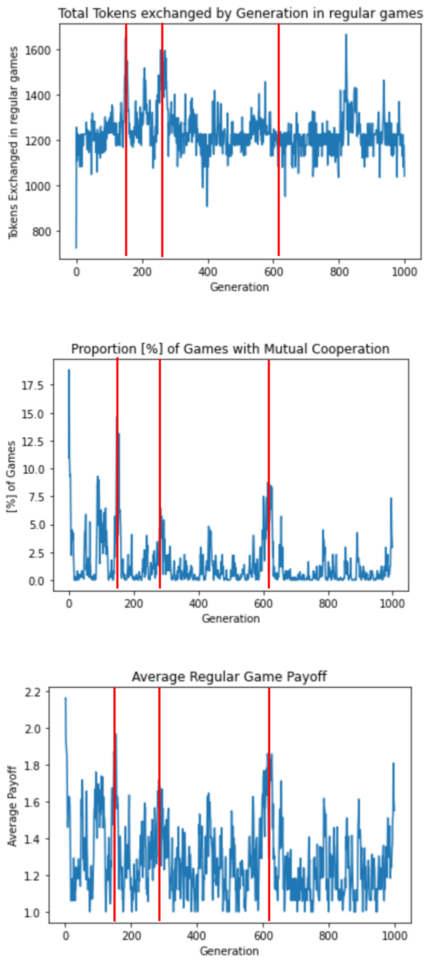

Figure 6: The results of the prisoners' dilemma analysis

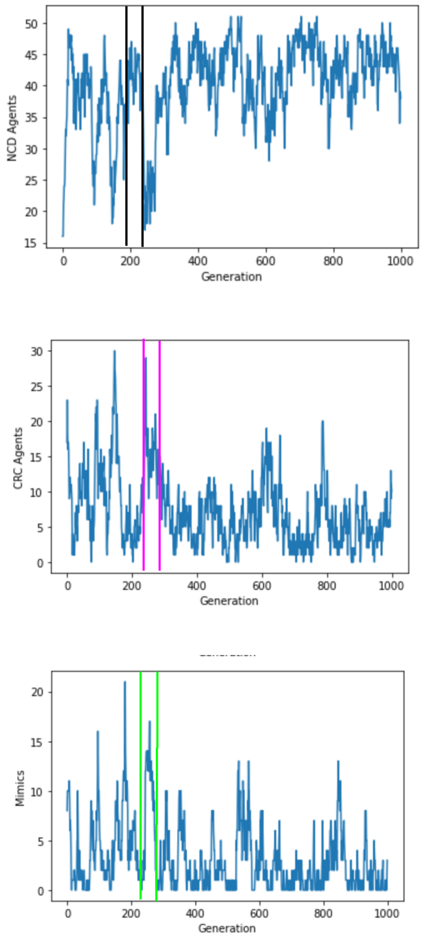

Figure 7: The results of the prisoners' dilemma analysis

4.1 Evolving Communication in the Prisoners' Dilemma

Due to how the Prisoners Dilemma is set up, with mutual defection being a dominant strategy in any case, we would expect the agents, whose goal is to maximize their own reward for each game, to play accordingly. The results show a wholly different picture though. As can be seen in Figure 6, have regular outbreaks of communication throughout the models' run time. It becomes clear that just before outbreaks of cooperative outcomes, there is a period of increased communication, made visible here via the total number of tokens exchanged. These cooperative outcomes then increase the average payoff for each agent for a short time, before communication subsides again.

J. Miller and Page, 2007 suggests the following macro-level dynamics for the agents and their strategic adaptation to explain these spikes in communication: Initially,

[3] https://aws.amazon.com/ec2/instance-types/

the system mostly consists of agents that don't communicate and immediately play 'Defect', this strategy we will henceforth refer to as *NCD* [4]. This leads to our initially expected view of continuous mutual defection. Eventually, a mutation will occur that gives an agent the following strategic pattern: initially it will send out a communication token, and, if it receives a communication token back, it will play 'Cooperate', otherwise it will play 'Defect'. This tactic can become more complex, depending on the number of states available to the agents, leading to so-called *handshakes* which allow agents to distinguish themselves as cooperative. This strategy will be referred to as *CRC* [5] These strategies don't immediately die out in the selection process since they receive at least as good a payoff as agents with the CD strategy profile. However, mutation will eventually lead to another class of strategic agents to emerge: the so-called *mimics*. They employ a similar strategy as the CRC agents, they send out the same communication token sequences as the CRC agents, but instead of cooperating, they will play 'Defect'. These kind of strategies will then receive a higher payoff against their cooperative counterparts and will thus continue to survive the selection process. Eventually, since the CRC agents will be driven towards extinction by their mimic counterparts, the system will become one of mutual defection again and the cycle starts anew.

The agent strategy analysis performed in Figure 7 supports such a dynamic within the context of the prisoners' dilemma, even though the phase transitions aren't as clear-cut as the above description suggests in the graphs shown, they certainly support it. For a full-fledged analysis of these dynamics, please refer to J. H. Miller et al., 1998.

At the core of these dynamics lies the dilemma that while the desired outcome may be beneficial to all agents, it is never truly achieved due to the emergence mimics and defectors.

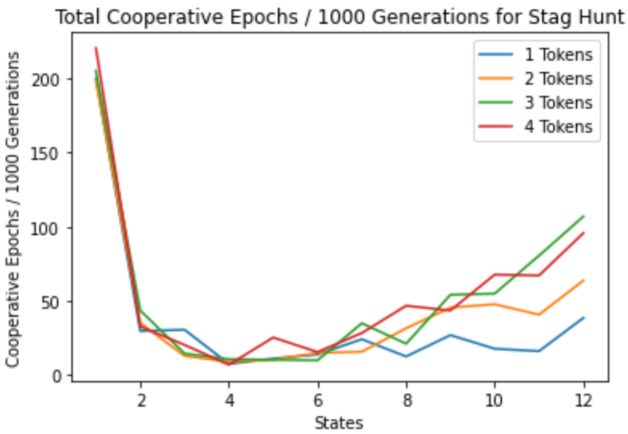

Figure 8: Generations with more than 10% (C, C) outcomes, averaged over 5 iterations of 1000 generations each

4.2 Impact of Exogenous Variables

In Figure 8 the impact of the states available to the agents for computation and the number of tokens available to the agents becomes evident. As the automata become more powerful, communication emerges as a tool to offset the core structure of the prisoners' dilemma. This effect is further strengthened by allowing the agents to communicate through increasingly more complex signals. Once the automatas are given six or more states for computation, it allows CRC agents to develop more intricate handshakes to identify other CRC agents and makes it harder for mimics to emerge, making the spikes in cooperation last longer. Even though the analysis performed could provide a clearer image if it was computed for a longer time and averaged over, say, 20 iterations, it is in line with the strategic evolution theory developed above and is supported by the analysis performed by J. H. Miller et al., 1998.

4.3 Evolving Communication in the Stag Hunt

For the Stag Hunt analysis, we are slightly extending the number of generations shown, putting $g = 1200$, due to the bigger scale at which dynamics occur in the Stag Hunt game. This will become clearer in the following analysis.

Similarly to what we saw previously for the Prisoners' Dilemma, there are spikes of cooperative, (S, S) outcomes, linked to the amount of communication occurring.

[4] No Communication and Defect
[5] Communicate and reciprocate communication

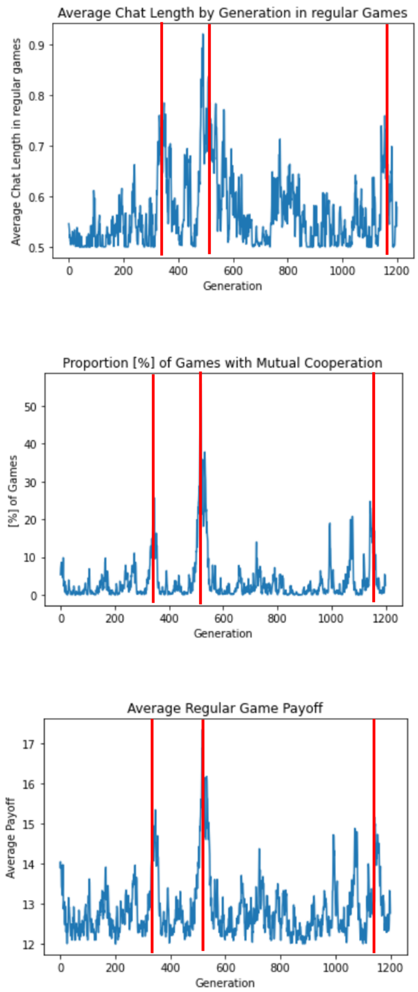

Figure 9: The results of the Stag Hunt analysis

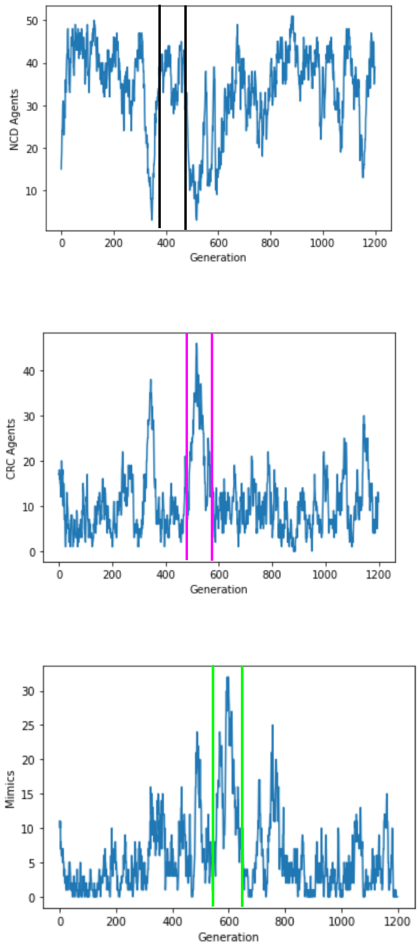

Figure 10: The results of the Stag Hunt analysis

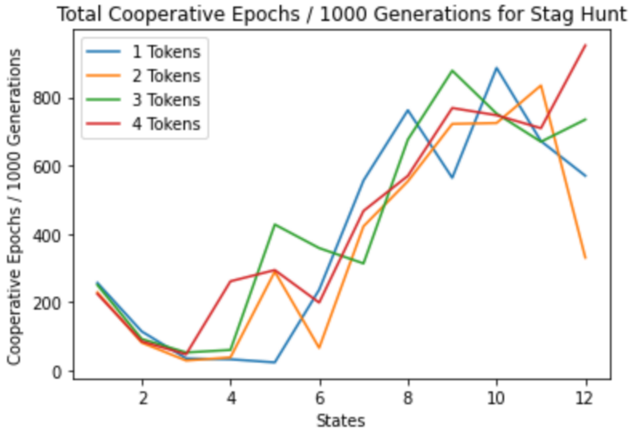

Figure 11: Generations with more than 10% (S, S) outcomes, averaged over 5 iterations of 1000 generations each

In this case, communication is shown via the average chat length for the respective regular games. However, while previously the highest spike in cooperation would not go above 16%, the scale at which our model spikes, is much larger, with the highest spike going above even 50%. This can, to a certain extent, be traced back to the nature of the Stag Hunt Game, for which (S, S) is a payoff dominant outcome. Still, similar dynamics seem to be at play here, with the agents being able to coordinate themselves via communication to achieve a higher payoff for both players.

Again, however, the emergence of counter-strategies is able to somewhat offset the positive effect communication has on the outcome [J. Miller and Moser, 2004]. The strategic dynamics of the agents are modeled in Figure 10. For the Stag Hunt, the NCD strategy is defined as an agent immediately playing 'Hare' without communication, the CRC strategy is defined as playing 'Hare' if the opponent doesn't reciprocate communication, but playing 'Stag' if he does and the mimic strategy is adjusted to play 'Hare' even after communication occurs. Another major difference between the two games for our theory of agent strategy dynamics is the length one 'cycle' in the previously defined framework takes. While the cycles and cooperative episode spikes in the Prisoners Dilemma were rather short lived (the highlighted cycle lasted for about 100 generations), the dynamics here seem to suggest longer cycle durations, with the highlighted cycle here lasting well over 200 generations.

This, allows us to infer that in the context of a coordination game, such as Stag Hunt, communication can allow agents to maintain desired outcomes over a longer period of time, allowing for reliably higher payoffs [J. Miller and Moser, 2004].

In a similar fashion, giving the agents more complex automata as a base for their strategies and allowing more tokens for communication promotes stability and coordination in the Stag Hunt game, as can be seen in Figure 11.

5 Outlook

While the model we developed exhibits surprisingly complex dynamics at the macro-level, that provide valuable insights into the development of very successful strategies through the use of communication, we are unable to make inferences about the endogenous development of languages in such a context. Recent advancements in the field of AI, especially Neural Networks, who exhibit similar macro-level features as the genetic algorithm used for agent adaptation in this model [J. Miller and Page, 2007], allow for a much deeper study into the emergence of grammar in agent-based models. One example of such a study can be found in Steels, 2016. However, while with increasing computational performance we could eventually model a system of agents on a geographical maps that develops their own grammar, language and maybe even hints of local (sub-)cultures at some point we might have to pose ourselves the question of even if we can, whether we should, since such a model at a certain no longer provides insights that are applicable to our real world, which would go against the very intention of agent based modeling.

Bibliography

Darwin, C. (1859). *On the origin of species by means of natural selection* [or the Preservation of Favored Races in the Struggle for Life]. Murray.

Frantz, D. R. (1972). *Non-linearities in genetic adaptive search* (tech. rep.).

Goldberg, D. E. & Holland, J. H. (1988). Genetic algorithms and machine learning.

Miller, J. & Moser, S. (2004). Communication and coordination. *Complexity*, *9*, 31–40. https://doi.org/10.1002/cplx.20034

Miller, J. & Page, S. (2007). Complex adaptive system. an introduction to computational models of social life. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*, 178–199.

Miller, J. H. (1996). The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior Organization*, *29*(1), 87–112. https://doi.org/https://doi.org/10.1016/0167-2681(95)00052-6

Miller, J. H., Butts, C. & Rode, D. (1998). Communication and cooperation.

Moore., E. F. (1958). Gedanken-experiments on sequential machines. automata studies, edited by c. e. shannon and j. mccarthy, annals of mathematics studies no. 34, litho-printed, princeton university press, princeton1956, pp. 129–153. *Journal of Symbolic Logic*, *23*(1), 60–60. https://doi.org/10.2307/2964500

Rousseau, J.-J. & Cranston, M. W. (1984). *A discourse on inequality / jean-jacques rousseau ; translated with an introduction and notes by maurice cranston*. Penguin Books Harmondsworth, Middlesex, England ; New York, N.Y., U.S.A.

Steels, L. (2016). Agent-based models for the emergence and evolution of grammar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *371*(1701), 20150447.

Wikipedia contributors. (2004). Plagiarism — Wikipedia, the free encyclopedia [[Online;]]. https://en.wikipedia.org/wiki/Communication

Statement of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Zurich, 28th June 2021

Zermin, Adrian