Project Proposal

Natural Computing

Stijn Boosman	*s1018438*
Florian Handke	*s1017134*
Mart van 't Spijker	*s1018582*
Radboud University	*May 7, 2023*

1 The Problem Statement

Non-cooperative repeated two-player games have been a topic of study for almost a century now. The mathematical framework that was developed to study such scenarios, coined Game Theory, has proven useful in a variety of research fields, including biology (population dynamics), politics (negotiation), and computer science (resource allocation), just to name a few. In these contexts, dominant strategies play an essential role as they represent an action that maximizes a "player's" outcome independently of the actions of other players which can lead to either desirable or undesirable states of a system that cannot be resolved easily. Although they exist in the original version of the so-called Prisoner's Dilemma, which is considered the classic example of non-cooperative two-player games, they do not manifest in the iterative multiplayer version of the game. That is, a population of N players matched M times at random to play the game for a fixed number of rounds K which is unknown to them. There are, however, so-called Nash equilibria, which are states of a system (i.e., the population of players) in which effective strategies can coexist while no player has a short-term incentive to deviate from their chosen strategy. The emergence of such equilibria depends on the initial conditions, for example, reward structure, a player's capabilities (memory and reasoning), and the discounted value of future rewards. Our project aims to investigate the effect that limited memory might have on the evolution of effective strategies and Nash equilibria in populations of varying sizes. If time allows, we would like to moreover look into how social sanctioning, i.e., a different reward structure might affect population dynamics.

2 Methodology

In order to simulate the repeated multiplayer version of the Prisoner's Dilemma such that it has the ability to potentially evolve effective strategies we will be needing the following components:

- **The players**: As mentioned before, we will need players capable of remembering their current opponent's actions, potentially even their previous opponents', as well as players who are capable of reasoning. This might be as simple as an *if-else*. These players will be modeled using a simple circuit, i.e., a multi-layer perceptron with a single hidden layer (f). Depending on the number of input units n the players thus gain the ability to reason based on the last n actions ($\mathbf{x}$) of their opponents. The underlying logic of such reasoning is represented by the perceptron's parametrization and activations. A player's action is then based on the output of the perceptron ($f(\mathbf{x})$) which encodes a value in [0,1] that then translates to action 1 if $f(\mathbf{x}) > 0.5$ and to action 2 if $f(\mathbf{x}) < 0.5$. An action is chosen at random if $f(\mathbf{x}) = 0.5$.
- **The ability to adapt:** For this we will be using a Genetic Algorithm (GA) that acts on the population of players (perceptrons). One generation comprises a population of N players who will be paired randomly with M other players from a non-exhaustive version of the same population. Each pairing equates to a repeated version of the Prisoner's Dilemma played for K rounds. The fitness of players is the average of their accumulated rewards over these sampled encounters. The genes of each player are represented by the respective perceptron's parameter values.

The concrete design of the GA is yet to be determined, i.e. mutation rate, cross-over, and parent selection. Furthermore, we will either need to design baseline simulations or use an existing baseline to compare our results to. One idea is to use the NEAT algorithm, which changes not only the weights of the perceptron but also the structure of its hidden layer(s). Such a baseline could potentially be provided by Axelrod's tournament (see below).

3 Related Work

NeuroEvolution of Augmenting Topologies (NEAT)[1] is a genetic algorithm that alters both the weights and the topology of a network. On simple tasks, NEAT can be more effective in finding a solution than other similar algorithms.

In his book 'The Evolution of Cooperation', Robert Axelrod investigates the most effective strategies in a long-running tournament where different kinds of computer programs with practically infinite memory play the iterative multiplayer version of the Prisoner's Dilemma. Throughout repeated tournaments, he finds that in an equilibrium state, certain types of strategies are more effective than others. One such strategy is the 'Tit-for-Tat' strategy. A strategy that only uses the last action of its opponent to conclude its next action [2]. The class of strategies that were deemed to be effective could potentially serve as a baseline for our experiments as they emerged from a setup in which players were given practically infinite memory.

Evolving systems that play the Prisoner's Dilemma have been extensively studied as done in a paper by David M. Chess [3]. In his work, the individuals playing the game are represented as arithmetic expressions with three variables. He finds that one can distinguish between roughly four stages throughout multiple generations: The era of Exploitation (domination of defectors), the Nadir (mostly interactions between defectors), the Growth of Trust (rational entities begin to multiply), and the last stage Equilibrium (cooperation is the rule). Entities in the Equilibrium are mostly Tit-for-Tat players and Nice-but-Unforgiving players.

References

[1] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[2] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *science*, vol. 211, no. 4489, pp. 1390–1396, 1981.

[3] D. M. Chess, "Simulating the evolution of behavior: The iterated prisoners' dilemma problem," *Complex systems*, vol. 2, no. 6, pp. 663–670, 1988.