Simulating Economic Learning in Dynamic Strategic Scenarios with a Genetic Algorithm

Vinícius Ferraz* Thomas Pitz†

March 2, 2021

Abstract

The application of game theory features in computational environments has been extensively per- formed in distinct fields, as the desire to reproduce strategic behaviour applied to solving complex optimization problems is part of a growing niche in artificial intelligence. This paper introduces an experimental and exploratory approach, combining game theory and Genetic Algorithms to create a model that simulates social interaction situations, economic learning and biological evolution. The introduced construct aimed at allowing for the evaluation of how players change their strategies over time, towards an optimal outcome. Specific 2×2 strategic form games as objects of analysis and three distinct strategy selection rules (the Nash equilibrium, Hurwicz Criterion and a Random method) were used in this study. Such games act as strategic scenarios simulations and are treated as individuals of a genetic population, from which they are evaluated, selected and reproduced. The outcome indicated an optimized player environment, demonstrating the maximization of the overall individual payoffs, while transforming the games in such a way that agents can coordinate their strategies and achieve an Evolutionary Equilibrium with optimal payoffs. Additionally, the mutation of the populations into a robust set of fewer isomorphic games featuring the strong characteristics from well-performing individuals is observed, as a consequent to the evolutionary learning process.

Keywords: Game theory, agent-based modeling, simulation, Genetic Algorithms, economic learning, artificial intelligence

*Alfred-Weber Institute of Economics, University of Heidelberg

†Faculty of Society and Economics, Rhine-Waal University of Applied Sciences

1 Introduction

Game theory models, in particular the strategic or normal form of games, are used to describe strategic scenarios (Morgenstern and Von Neumann, 1953). Among the simplest models are 2×2 matrix games, modelling fixed strategic scenarios in which the set of players, their strategies and payoff functions do not change. Even if occasionally these games are considered over several rounds, the overall strategic framework remains constant. In the present paper, we describe an exploratory and experimental approach that allows us to simulate the transformation of the payoff matrix in 2×2 games by using Genetic Algorithms. Over several periods, simulated agents play the 2×2 games. After each period the played games were evaluated by the agents depending on their experience in the past rounds, i.e. their received payoff. A Genetic Algorithm changes the games and provides a modified scenario for the players. We describe this process depending on different strategy selection algorithms and different pools of 2×2 games.

In the reality we live in, as society progresses, humanity is constantly going through systemic transformations in many aspects (micro- and macro-scale) on a global level. Technological and scientific advancements, together with changes in economic, political, sociological and other strategic scenarios lead to new decision environments. Consequently, in such emerging decision environments where strategic interaction takes place, this forms the basis to create the need for individuals to update their beliefs and adapt to the new informational and strategic structures, as argued by Freedman (2017), who cited the example of socioeconomic and political transformations.

Agent-based models of society introduce a rich analysis scenario of autonomous agents that have the capacity to interact with each other and with the environment in which they are inserted, according to predefined rules of behaviour (Billari et al., 2006). Through this manner, one can understand the properties of complex social and economics systems (enabled by simulation analyses), which can be especially useful when performing experiments, as stated by Axelrod (1997). This method has seen a growing application in multidisciplinary contexts in the last few years, as it can be instrumental to handle complex interactions and uncover emergent behaviour (Bonabeau, 2002). One of the important research capabilities of agent-based simulations is how to represent human actions in real-world scenarios. Building up from the neoclassic assumption of rational behaviour, the study of strategic interaction in the field of game theory provides a formal model for analysing individual self-maximizing decision-making behaviour in a spectrum of varying hypothetical scenarios (Morgenstern and Von Neumann, 1953). The widely discussed rationality paradigm assumes that individuals know their preferences and make the best possible decisions based on the available information. This model of behaviour enables the reproduction of decision-making features in computational environments, where virtual agents interact according to rules, allowing for the investigation of complex social patterns (Billari et al., 2006)

In order to capture the dynamic effects of repeated interactions, the field of evolutionary game theory provides a toolkit for the analysis of games played by socially conditioned players over time. This is in contrast to the standard interpretation of game theory, which analyses static situations. This branch of the theory combines economics and biology fields, modelling dynamic and strategic interaction with tools that resemble biological evolution (such as natural selection, breeding and mutation) (Gintis et al., 2000). The evolutionary approach introduces the notion of pre-defined strategies that are repeatedly applied in an evolutionary process and operating dynamically on the distribution of behavioursm, as stated in Weibull (1997).

These fundamental behavioural changes can, over time, be observed in empirical scenarios since performing decisions can have an effect not only on the strategic behaviour of the agents involved, though they can also influence environmental conditions and individual preferences. As an example, Heckathorn (1996) applied the dynamic interaction logic to the well-known public goods dilemma, where changes in variables such as the production function, individual valuation, cost structures (and other influencing factors) can transform the Prisoner's Dilemma situation into different games, providing four examples of such game transformations: Chicken, Assurance, Altruist's Dilemma and Privileged. Complementarily, Simpson (2004) presents empirical evidence on how social preferences can transform a Prisoner's Dilemma into a game of Assurance (situation also approached in Hayashi et al. (1999); Kollock (1998)).

With the aim of reproducing economic learning behaviour applied to solving complex decision optimization problems arising from repeated interaction, this paper introduces a simulation model for adaptive learning and optimization under constraints, through the implementation of an evolutionary mechanism. This mechanism is denominated as the concept of Genetic Algorithms, defined as a metaheuristic inspired in the processes of biological evolution, with the objective of solving computational problems (combinatorial search and optimization) and modelling evolutionary systems (Holland, 1992; Goldberg and Holland, 1988).

2 Economic Models of Learning

As Camerer (2003) outlined, sometimes the economic theory neglects aspects of learning. If there is perfect information and rationality is assumed, the equilibrium point will always be known from the start and individuals will only change the equilibrium once information changes. Moreover, Camerer and Weigelt (1988) emphasize the importance for achieving better outcomes in experimental games, especially when dealing with scenarios having potentially inefficient equilibrium outcomes, such as trust games, public goods games, beauty contests and others. Consequently, well formulated economic learning theories are important - in the sense that they provide predictive power, coherence, and concomitantly reveal new insights (Camerer, 2011).

Economic learning models, in contrast to the neoclassic assumption of perfect rationality, has developed a growing interest by decision behaviour scholars, such as Baddeley (2018); Dhami (2016), where the authors summarize and categorize learning the most discussed learning models into three basic models. Belief Learning (BL) models describe actions built by distributing probabilities to each strategy combination from other players, yielding a time-dependent belief function, in which beliefs about other player's actions are updated as the situation unfolds, as a strategic process of anticipating actions of others and acting accordingly (Cheung and Friedman, 1997). Reinforcement Learning (RL) models are defined as models that draw insights from classical Pavlovian-based behavioural psychology experiments (Pavlov, 2010), where the process of response to rewards drives the learning in sequential decision processes through repetitive experiences, as described in Baddeley (2018) (see Harley (1981) for an early application). The Experienced Weighted Attraction (EWA) model introduces a hybrid model encompassing elements from BL and RL models (Camerer and Hua Ho, 1999) in order to provide a method with higher predictive and descriptive power, where the experimental evidence demonstrates the potential of capturing more (sophisticated) players, empirically assessed by Chmura et al. (2012); Brunner et al. (2010); Pangallo et al. (2020). Recent evolutionary approaches to learning have been applied in conjunction with economics for simulation of learning behaviour, associated with a broader concept of learning. This concept was previously studied in depth as evolutionary game theory, where models assume that agents choose their strategies through an adaptive process in which they progressively discover which strategies are preferred, given the scenario (Weibull, 1997; Samuelson, 1997). The latter concept is explored more in detail here.

2.1 Evolution-Inspired Computational Models

The multidisciplinary combination of game theory and genetic programming has been growing in many distinct fields, ranging from economics and sociology to computer science and natural sciences such as biology. Evolutionary game dynamics provide comprehensive frameworks for studying interaction, learning and evolution (Roca et al., 2009). In addition, economic models of learning (including evolutionary models), in contrast with the neoclassic assumption of perfect rationality, allow for the possibility to study individuals (or computer agents) as they can learn and update their beliefs in light of new information, since the application of an evolutionary model assumes that strategies can change over time (Baddeley, 2018). According to Axelrod et al. (1987), when interacting in complex environments, individuals are not fully able to analyse the situation and consequently calculate optimal strategies. Alternatively, the strategies are adapted over time and based upon achieved results, demonstrating how a Genetic Algorithm can be particularly adept as a learning mechanism for evolving sophisticated and effective strategies. The given approach serves as an inspiration for the analysis performed in this article. Similar methodologies are outlined as follows.

Isaac (2008) provided an introductory overview of an agent-based model with Genetic Algorithms in the iterated Prisoner's Dilemma, reporting variations in the payoff structures that creates new player types, introducing an interaction between payoff cardinality and players' attributes. Pitz et al. (2005); Chmura et al. (2007), presented a novel simulation model for analysing action patterns in social systems that are mainly based on the concepts of Genetic Algorithms and the Theory of Action Trees (Goldman, 1971). They explain how the emergence and disappearance of actions can be described with a uniform algorithm, succeeding in endogenously eliciting comprehensive changes in the agents' behaviour.

Similarly, the gameEA algorithm for optimization problems reported in Yang (2017) builds up on a similar conceptual framework, introducing a game theory-inspired evolutionary model that updates the strategy sets by replacing individuals of the population with better performing offspring, generated by imitation or belief learning operators, therefore creating a model that outperforms four other algorithms (StGA, IMGA, RTS and DPGA) often used for similar purposes. Pereira et al. (2020) also introduced a constrained optimization model that explored two ideas, the first being a Genetic Algorithm with social interaction (for diversification of solutions in the selection process). The second model consisted of game-based crossovers (tournament simulations for more diverse offspring). The presented construct demonstrated robust performance, when compared to traditional methods, in engineering design optimization process, consequently highlighting the benefits of combining Genetic Algorithms and game theory.

Another similar framework was developed by Gooding (2014), who formulated a simulation model for capturing evolutionary trends observed in society, such as wealth aggregation, inequality and climate change. Such a study depicted that where experimental evidence confirms changes in actions, environments and decision making, such social trends remain robust and resistant to change, providing insights in how to influence social progress. In a somewhat similar context, Glynatsi et al. (2018) used an evolutionary game theoretic model in the ecology field to examine the interaction between poachers and wildlife. The model analysis reports how devaluation of rhino horns would likely lead to higher poaching activity and that such strategy is only effective when applied together with disincentives, aiming to contribute to informing the discussion about debates in the topic with empirical evidence.

Other interesting applications of the combination of Genetic Algorithms and game theory described in published literature include practitioners in other distinct and diverse fields, such as engineering (Périaux et al., 2001), energy ((Castillo and Dorao, 2012, 2013; Mohamed and Koivo, 2011), communications (Kusyk et al., 2011), land usage (Liu et al., 2015), biology (Hamblin and Hurd, 2007) and ecology (Hamblin, 2013).

3 Applied Game Theory for Modelling Behavior

The widely studied field of game theory, first described in Morgenstern and Von Neumann (1953), introduces mathematical models of strategic interaction between rational decision-makers. The so-called games in the theory act as simplified representations of real-world scenarios, providing normative and descriptive prediction power regarding the outcomes of the strategic situation involving self-interested, utility maximizing individuals (Samuelson, 1997). Aiming to understand human behaviour in strategic situations, specialists such as Camerer (2011) have applied concepts of psychology and game theory in a wide range of experiments to further develop models that can give us insights on how people behave in the real-world.

Furthermore to the above, the evolutionary niche of the theory provides a framework that is tuned for understanding the selective forces affecting the evolution of strategies of agents involved in dynamic strategic interaction. This provides a more effective method to predicting evolutionary outcomes, in comparison to standard mathematical models (Adami et al., 2016). Friedman (1991) broadly defines evolutionary games as models of strategic interaction over time, in which higher yielding strategies tend to displace inferior strategies, where there exists a degree of inertia and players do not systematically attempt to influence others' future actions. The term also presupposes the survival of the fittest mechanics and that overall behaviour does not change abruptly. Since this article proposes to simulate how learning can change the strategic scenarios by influencing the prevalence of higher payoffs, evolutionary games provide an appropriate theoretical basis for the proposed analysis.

This article adopted the standard representation of strategic form games as a model of simultaneous interaction between two agents, which is denoted by a 2×2 matrix, encompassing the following elements: the (two) players, who are the parties making the decisions; the strategies that can be selected by each player (two for each) and the payoffs being the rewards received as a function of the chosen strategy (Morgenstern and Von Neumann, 1953; Robinson and Goforth, 2005). This representation of games focuses on static analyses, ignoring dynamic aspects such as the order of the moves of the players, change moves and the informational structure. This approach suggests which strategies are more likely to be played by each player, or alternatively recommends players which strategies to choose in similar scenarios (Maschler et al., 2013). In Robinson and Goforth (2005)'s notation, the players in the context of this article are named after how the strategy profiles are organized, with the player ROW's strategies displayed the rows of the matrix and player COL's strategies in the columns, respectively.

The class of 2×2 strategic form games processed by this simulation model is based

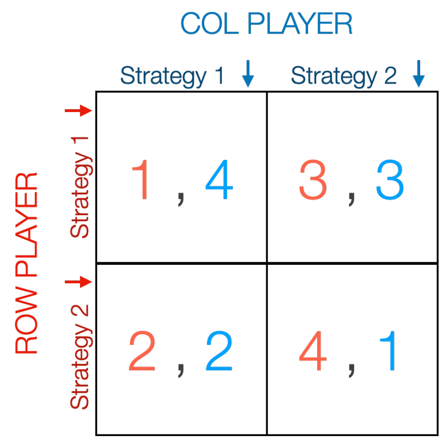

Figure 1: Standard representation of Prisoner's Dilemma, based on Robinson and Goforth (2005). Players ROW and COL have a set of two pure strategies each, in this case denoted as 1 and 2. The first number in each cell of the matrix stands for player ROW's payoff for the corresponding strategy profile, whereas the second represents player COL, respectively. In summary, the numbers in the matrices represent the payoffs of pure strategies, given the actions of the players. A player can also have mixed strategies, that is - a probability distribution over pure strategies (Fudenberg and Tirole, 1991; Robinson and Goforth, 2005)

on the "periodic table" categorization provided in Robinson and Goforth (2005), which formally connects all ordinal rank games with distinct player preferences, since the games are topologically linked by swaps in adjoining payoffs. The space of 2×2 is infinite, though as we are only interested in ordinal preference relations, we can concentrate on classes of isomorphic games, where for each class we can choose one representation of the form $\{1, 2, 3, 4\} \times \{1, 2, 3, 4\}$. Respectively, as Robinson and Goforth (2005) have demonstrated, there are 576 ways to arrange two sequences of four numbers in a bi-matrix scheme. Since the ordinal structure of a game does not change by switching rows, columns or both simultaneously, the 576 games can be reduced by a factor of 4, to 144. For the analysis formulated in this article, all 144 unique classes of games - which represent a wide range of well-known applied game theory situations such as: the Prisoner's Dilemma, Chicken game, Stag Hunt, Battle of Sexes and many others - will be considered in the simulation model for a broader representation of strategic scenarios. In complement to the periodic table approach, Bruns (2010) further categorized the games by similarity, adding another layer of information to the analysis.

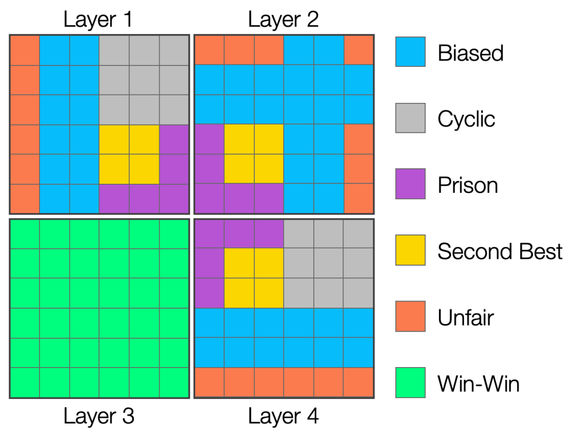

Figure 2: Representation of the Game Families and Layers in the Periodic Table of 2×2 games (Bruns, 2010, 2011, 2015a,b). This reveals families of games that are similar to each other, where individuals share similar characteristics such as the equilibria and payoff structures. The game family categorization, together with the layer categorization from Robinson and Goforth (2005), is the criteria used to split the list of games into distinct populations using the families' representation to create game pools with similar characteristics while the layers' representation (except for layer 3, which coincides with the win-win games family), allow for gene pools with more diverse characteristics.

Game Family	**Nash Equilibria**	**Pareto-optimal**	**Dominant Strategies**	**Count**	**%**	**Details**
Biased	1	2	0-2	44	31%	One player gets the best and the other second best outcome
Cyclic	0	2-4	0	18	13%	In each cell one player would prefer to change their move, no pure strategy equilibrium
Prison	1	2-3	1-2	15	10%	Dominant strategies based on individual incentives leads to a worse outcome than cooperation
Second Best	1	3	1-2	12	8%	Both players gets the second best outcome
Unfair	1	2-3	1-2	19	13%	One player gets the best and the other the second-worst outcome
Win-Win	1	1	1-2	36	25%	Both players get the best outcome

Table 1: Formal Description of the Game Families (Bruns, 2015a). The core idea of the periodic table concept remains the same in this representation, focusing on strategic situations where the outcome of decisions is dependent from other decisions within the designated scenarios, as the payoff structures (scenarios) can lead to win-win situations or tragic failures, when individual incentives clash with strategies that could benefit both players. Utilizing the periodic table as a holistic view with a diversity of representations of strategic situations, one can notice that changes in payoffs can turn one game into another, potentially shifting equilibrium outcomes and opportunities to totally different effects and consequences (Bruns 2015a, b).

4 Simulation Model Design

The core objective is to assess a Genetic Algorithm that allows the agents to change their given environments. These preferences are implicitly expressed by the fitness function, which is the measure applied to assess the quality of the selected strategies in terms of (expected) utility gained. The simulation process basically consists of playing the games in the designated populations based on three different strategy selection rules: (1) Nash equilibrium, (2) Hurwicz rule and (3) random selection. After the strategies are selected, the games are assigned fitness scores based on the aggregated payoffs from both agents' choices, so the games can be selected for reproduction in a way that favours better performing combined strategies. In the next step, the games are processed by the Genetic Algorithm, which selects two games (parents) from the pool, applies the crossover and mutation operators, recombining the binary sequences, thus creating a new game that inherits the parents' characteristics. The new games are inserted back in the population and the process is iterated for a fixed number of times. In order to manage complexity and processing requirements, the population numbers were kept constant throughout the iterations of the evolutionary model.

4.1 Populations and Data Structures

The population structure was based on a vector representation of the 2×2 games in strategic form, having the payoff structure represented by integer vectors (see Figure 4).

The analysis was rooted in two primary divisions of game pools based on families and layers. The first restriction was based on Robinson and Goforth (2005)"s division of layers in their definition of the periodic table of 2×2 games, which considers all the 36 neighbouring games according to the amount of payoff swaps, as depicted in figure 2. In this context, each of the four layers was taken as a separate population. This approach leads to a higher diversity in the games' structures during the learning process. The entire space (all 144 games, not allowing ties) was also handled as one distinct population, being processed apart

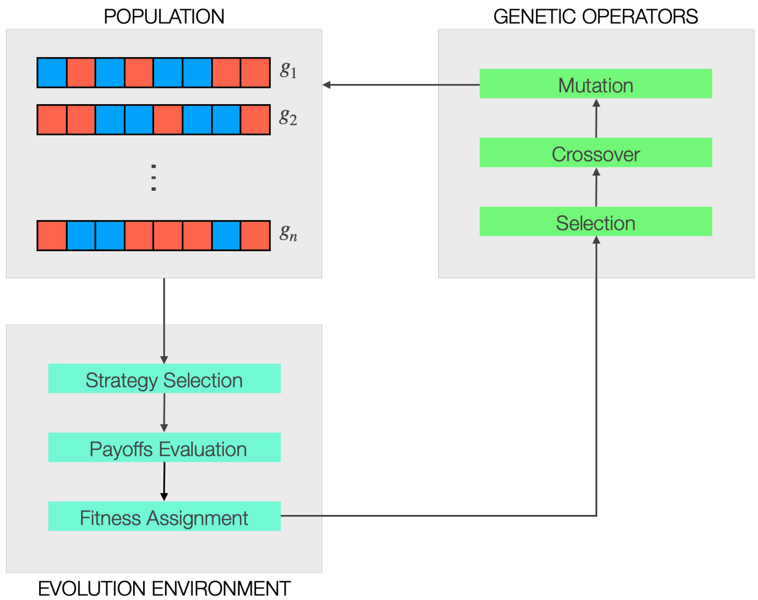

Figure 3: Simulation Model Overview. Firstly, the initialization of a populations of games. Secondly, a strategy selection and evaluation method based on the given decision rules that reads into the game vectors and selects strategies for the respective players, assigning the fitness scores based on the strategies' performance. The third step applies the genetic methods in the games' population, with the selection, crossover and mutation steps generating new individuals that are added to the population at the end of their respective generation.

from the others, allowing us to analyse the results of the population with a complete set of available characteristics in the pool. The second restriction was based on Bruns (2015a)'s game families categorization, based on their payoffs at Nash equilibria. The games were split into six families as described in table 1. Similar to the first restriction, each of the families was handled as separate populations of games so as to analyse how a pool with similar individuals develops throughout the generations towards the optimal point. These two restrictions separated our group of games into eleven distinct populations, processed and analysed individually.

4.1.1 Binary Encoding

When the games were ready to be processed by the Genetic Algorithm, a binary representation of the vector form was adopted, transforming the games into a collection of bits

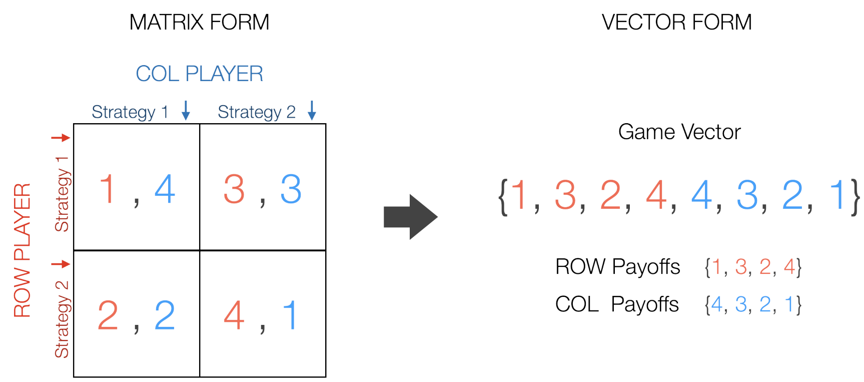

Figure 4: Game Vector Encoding Scheme. The population structure was based on a vector representation of the 2×2 games in strategic form, having the payoff structure represented by integer vectors. The conversion logic applied to the entire topology as well as the possible transformations and new isomorphic games that might be created during the evolutionary process. This outlines how the Prisoner's Dilemma game was encoded into a vector representation for this simulation.

(represented by 0 or 1) as the adopted classes of games here are represented by an integer between1 and 4 ($\{1, 2, 3, 4\} \times \{1, 2, 3, 4\}$). Each pair of bits encoded one of the four possible integer numbers that defines the payoff structures in the games. Consequently, the binary encoding was based on four possible sequences, using a double binary representation: $0, 0 \rightarrow 1$; $0, 1 \rightarrow 2$; $1, 0 \rightarrow 3$; $1, 1 \rightarrow 4$.

Following this, as an example, the binary representation of the Prisoners' Dilemma vector form would be given as: $[1, 3, 2, 4, 4, 3, 2, 1] \rightarrow [0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0]$. All game vectors contained 16 bits, with each pair of bits representing one of the payoffs in the matrix. The game vectors were only allowed to be changed following this structure, so the adopted structural scheme remained the same.

4.2 Evolution Environment: Strategy Selection and Fitness

This chapter describes the mechanisms applied for the agents' strategy selection rules, defining how the players should decide in the introduced scenarios. In order to obtain enhanced exploratory overview on how a decision affects these scenarios, three distinct methods for selecting strategies were adopted

4.2.1 Nash Equilibrium Strategy Selector

According to the standard notation in the literature, having 2×2 games in the centre of this analysis, allowed (A, B) to be a bimatrix game, where A and B are 2×2 matrices of payoffs belonging to the ROW player 1 and COL player 2, respectively. In strategic form games, the decisions are made simultaneously, that is, the choice of a row i by player 1 (or ROW) and column j by player 2 (or COL) are performed at the same time. Based on the outcome of the strategies, the players receive a payoff a_{ij} and b_{ij}. The payoffs represent risk-neutral utilities. In this case, when facing a probability distribution, the players will naturally aim to maximize their expected payoffs (von Stengel, 2007). Based on the taxonomy of the class of games selected for this study, A and B were be assumed to be represented by 2×2 matrices containing payoffs ranging from 1 to 4 (as in Bruns (2010, 2011, 2015a,b); Robinson and Goforth (2005)).

The Nash equilibrium can be defined as a strategy profile in which every strategy is an optimal response to other players' strategies. This logic can be applied to pure and mixed-strategy (probability distribution over the available choices) profiles. A strategy profile σ^* is considered a Nash equilibrium if, for all players, i, the following conditions are fulfilled

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \text{ for all } s_i \in S_i. \tag{1}$$

As expected, utilities are linear in terms of probabilities. If a non-degenerate mixed strategy is utilized by a player in a Nash equilibrium, it must be indifferent between all other pure strategies assigned with a positive probability. A strict Nash equilibrium exists when each player has a unique best response to its rivals' strategies. In this case, s^* in considered a strict equilibrium if it is also a Nash equilibrium, for all i and all $s_i \neq s_i^*$:

$$u_i(s_i^*, s_{-i}^*) > u_i(s_i, s_{-i}^*). \tag{2}$$

A strict equilibrium is necessarily a pure-strategy equilibrium. This form of equilibria remains strict when the payoff functions are slightly perturbed, as the strict inequalities remain satisfied. This concept can be thought as a consistent means to predict the way the games will be played, since in the occurrence of a Nash condition, no player has any incentives to play differently. In contrast, the prediction that any fixed non-Nash profile will take place implies that at least one player will make a mistake, either in its prediction of the opponents' action or - given that prediction - in the optimization of its own payoff. Any Nash-equilibrium strategy profile must focus only on strategies that are not strictly dominated - a player's payoff could be increased by replacing a dominated strategy with another one that dominates it. However, it is important to point out that Nash equilibria may also assign positive probabilities to weakly dominated strategies. Another critical remark on this concept is that many games have more than one Nash equilibria. In this case, the assumption that a Nash equilibrium would be played, relied on the existence of some mechanism or process that led all players to expect the same equilibrium (Morgenstern and Von Neumann, 1953; Fudenberg and Tirole, 1991; Maschler et al., 2013).

The method adopted for computing the Nash equilibrium was based on the support enumeration concept, which calculates all possible Nash equilibria in bimatrix games (von Stengel, 2007; Widger and Grosu, 2008). An open-source library of this method, provided by Knight and Campbell (2018), was adopted for performing the calculations which return all equilibria for a degenerate 2×2 game $(A, B) \in \mathbb{R}^{m \times n^2}$, for all $1 \leq k_1 \leq m$ and $1 \leq k_2 \leq n$ (enumeration of all possible equilibrium strategies); for all pairs of support, (I, J) with $|I| = k_1$ and $|J| = k_2$. This was, in essence, finding the support of strategies which are played with a non-zero probability. At this point, the algorithm evaluated the best response condition, ensuring no better utility resided outside of the supports.

The steps outlined in equations (3) and (4) are repeated for all potential support pairs.

$$\begin{aligned} &\sum_{i \in I} \sigma_{ri} B_{ij} = v \text{ for all } j \in J, \\ &\sum_{j \in J} A_{ij} \sigma_{cj} = u \text{ for all } i \in I. \end{aligned} \tag{3}$$

Sequentially, for considering mixed strategies:

$$\begin{aligned} &\sum_{i=1}^{m} \sigma_{ri} = 1 \text{ and } \sigma_{ri} \geq 0 \text{ for all } i, \\ &\sum_{i=1}^{n} \sigma_{ci} = 1 \text{ and } \sigma_{cj} \geq 0 \text{ for all } j. \end{aligned} \tag{4}$$

For games with more than one Nash equilibrium, there were two equilibrium-selection rules. One rule based on the payoff dominance concept formulated by Harsanyi et al. (1988) for simulating maximizing behaviour. For each game, there was at least one equilibrium (pure or mixed), and we could have finite or infinite equilibrium points. In our structure, three main configurations were identified: pure-strict, pure-weak (when a stronger pure-strategy equilibrium was available) or mixed. A Nash equilibrium is payoff dominant, and henceforth selected, if it is Pareto superior to all other possible equilibrium situations in a game, meaning that agents would agree in choosing this equilibrium as it offers at least as much utility as the other identified equilibria. The second rule applied a simple random selection from the sample of computed Nash equilibria for each game, enabling the comparison of the maximization versus randomization approaches, as well as for measuring the effects in the evolutionary process generated by more diverse approaches to Equilibrium selection.

4.2.2 Hurwicz rule Strategy Selector

Distinctively, the second strategy selection rule adopted in this study was based on the Hurwicz criterion (Hurwicz, 1951). This rule introduced a coefficient of realism, α (also symbolizing pessimism or optimism, depending on whether its emphasis was placed in the coefficient best or worst possible outcomes). This method provided itself as a tool for balancing pessimism and optimism in decision-making under uncertainty scenarios, allowing decision-makers to account for different possible outcomes. The pessimistic option is based on the *maximin* criterion, while the optimistic option is based on the *maximax* criterion. The α parameter introduces a weighting factor between both extremes, simulating different degrees between behavior profiles.

The method, based on Gaspars-Wieloch (2014)'s explanation, applies the following formula.

$$h_j = \alpha \cdot w_j + (1 - \alpha) \cdot m_j, \tag{5}$$

resulting in the Hurwicz's criterion, h_j, with α as the coefficient of realism, being $\alpha \in [0, 1]$. In this paper, 0 represents the pessimistic extreme, that is the risk-averse behavior, while 1 represents the optimistic extreme, or the risk-prone behavior (Colman, 2016). The optimal alternative between the two is satisfied by:

$$h_j = \max_j\{h_j\} = \max_j\{\alpha \cdot w_j + (1 - \alpha) \cdot m_j\}. \tag{6}$$

In this study, three variations of the Hurwicz coefficient (α) were applied, simulating three distinct decision-making profiles: pessimistic (0.0), neutral (0.5) and optimistic (1.0). This strategy selector introduced another model of behaviour profiles and enriched the dataset for the analysis of simulation results. Other applications of the Hurwicz criterion in decision-making under uncertainty scenarios include see Jaffray et al. (2007), Pažek and Rozman (2009) and Puerto et al. (2000).

4.2.3 Random Strategy Selector

The third strategy selection mechanism consisted of a purely random choice of strategies for both players. This method was added mainly as a control scheme, so one could assess if the populations were able to progress in terms of utility by not having any simulated decision rule and only relying on the maximization mechanism of the Genetic Algorithm - through selection, crossover and mutation.

4.2.4 Fitness

The fitness function measured the quality of the strategies selected. With the possibility of instructing the algorithm on what is considered a good or bad solution, the practitioner had the ability to influence the design choices for fitness function, creating guidance for the search process. In case of the simultaneous optimization of multiple objects, the fitness function values returned by each single object could be aggregated. An important aspect of this approach was a fair evaluation of the quality achieved, favoring high performing games while applying penalties to bad performance, as outlined in Kramer (2017).

Our fitness function reflected the players' preferences defined by the strategy selection rules applied at the gameplaying stage, consequently taking the aggregated payoffs from each player as the overall game utility. The previously defined strategy selectors returned an array of probabilities (pure or mixed) of the agents selecting between the available strategies. The method applied was inspired by the dynamic analysis of bimatrix games by Tanimoto (2015). Within the context of this study, we assumed a limited population with constant size (finite number of players per round). The previously defined strategy selectors returned an array of probabilites which would determine the outcome of the game played under those rules. In this case, the strategies adopted by an agent during the execution of a game scenario could be either *Strategy* 1 or *Strategy* 2, for which the probability distribution yielded were expressed by the following state vectors:

$$\begin{aligned} &Strategy\ 1; p_1 = (1 \quad 0), \\ &Strategy\ 2; p_2 = (0 \quad 1). \end{aligned} \tag{7}$$

Having the payoff matrix of the game structure represented as:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{8}$$

We could illustrate the dilemma and the decision on the hypothetical situation between two players, where player *ROW* adopted *Strategy* 2 and player *COL Strategy* 1 , described by the following matrix equation:

$$\pi_{2,1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} .(1 \quad 0) = C \tag{9}$$

The above scenario would define outcome of the game by the tuple of payoffs C, containing the realized payoffs (a_{ij}, b_{ij}) for players *ROW* and *COL*, respectively, which we will denote throughout the analysis as *utility*, for the sake of simplicity. The fitness (f) for a game (g) in the population (G) was then defined as:

$$f_g = (a_{ij} + b_{ij})^2. \tag{10}$$

The acquired utilities were aggregated up in order to represent the overall utility derived from a game, then squared to give higher weights to the best performing games in the game selection step, relative to the current population's performance. More details on the mechanics and effects generated by the nature of this fitness function in the Genetic Algorithm are outlined in the following chapters.

All variations of strategy selectors were applied to all game populations, as illustrated in figure 5. Once the strategies were selected and the utility scores were attributed to the entire set of games, the game population was ready to be processed by the Genetic Algorithm environment.

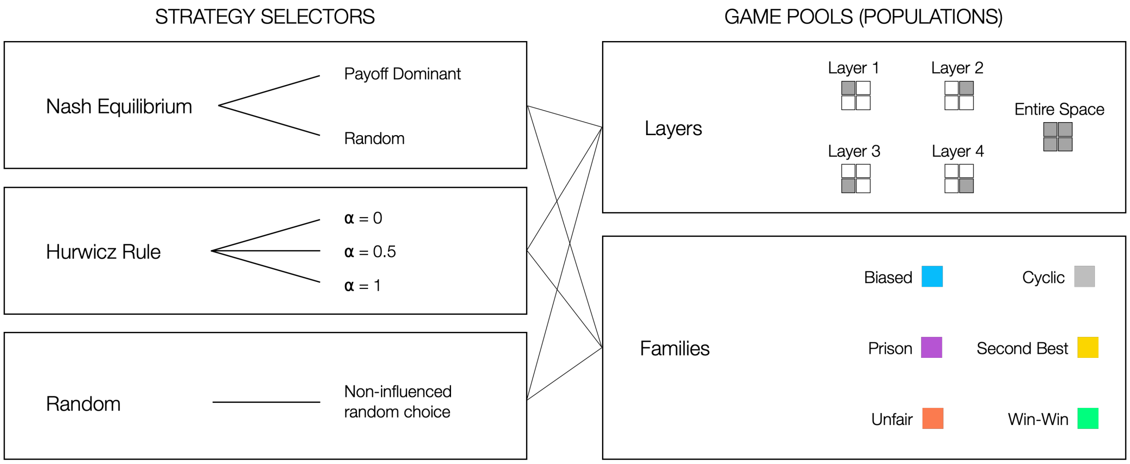

Figure 5: Overview - Strategy Selectors versus Populations. Each strategy selector and its modifications (such as the equilibrium selection rules for the Nash equilibrium or the different variations of the α parameter from the Hurwicz rule) were applied in each of the pools as outlined above, creating a complete matching round from all populations with all strategy selectors.

4.3 Genetic Algorithms Implementation: Genetic Operators

The application of the Genetic Algorithms method in this study was to model the collective learning process within a population of individuals. Each individual represented a search point in the space of all potential solutions for the introduced problems, as well as a potential temporal container of current knowledge regarding the laws of the environment created. The process started when a first population was initialized by an algorithm-dependent method and evolved towards enhanced performance regions of the search space by means of randomized processes of crossover (reproduction), mutation and selection, described as genetic operators (Back, 1996). The process of optimization in this context aimed to adjust the payoff structures in the games population, in order to extract the best outcomes for the defined strategy selection rules. The strategies performances, in terms of payoffs acquired, were taken as inputs for the fitness function. In this case, the population of games could evolve under specified conditions into a state that maximized the fitness, as specified by Haupt et al. (1998).

4.3.1 Selection and Replacement

Selection is a crucial factor for directing the searching process towards better individuals. According to Rowe (2015); Reeves and Rowe (2002), the fitness function assigns a positive score to the points in the search space where the optimal solution has the maximum (or minimum) fitness. Often, the fitness function is, in fact, the objective function of the problem in question. There are multiple selection methods defined in the literature, such as tournament, random, proportional among others. For a concise overview of the popular selection methods, see Back (1996) and Rowe (2015). The objective of the selection process was to use a non-deterministic approach, meaning that all individuals would be considered during the selection process. However, a totally stochastic method would lead to the inefficiency of the output solutions. In this case, a method that would fit the mentioned criteria was the fitness proportionate selection, where the probability of a game g being selected was based on its performance against the rest of the population: $\frac{f(g)}{T}$. Where f is the game's fitness scores and T is the total fitness of the current population (aggregated).

NEUTRAL VALUES

GAME	FITNESS
A	3
B	4
C	6
D	2
E	5
F	8

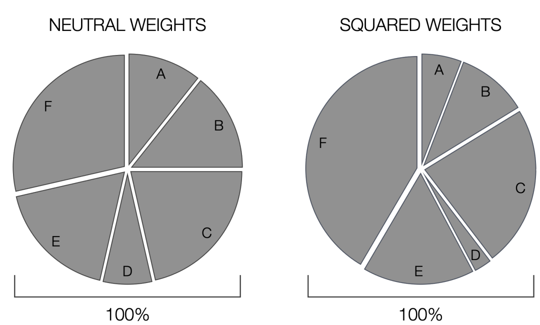

Figure 6: Fitness Proportionate Selection (Roulette Wheel) Overview. This approach consisted of a weighted version of a random choice, assigning a probability distribution over the population (where the most fit individuals received higher weights), hence having a greater chance of being selected. Since the range of our fitness figures was relatively narrow, the squaring of the fitness scores scaled the weights in order to obtain a distribution that favoured higher scores and penalized lower scores.

The first step of this method was to calculate relative fitness by dividing the individual rates by the total fitness figure for the entire population, producing a proportional weight to the chromosomes according to their performance. Since the range of the possible utility figures were small, if only the relative probabilities were used, the algorithm would select "bad" individuals more often, reducing its efficiency, hence the reason for using squared values.

The standard replacement rate chosen for the algorithm was 1, meaning that once the weights were assigned, the selection function performed a random selection over the population probability distribution and chose two individuals, becoming the parents, consequently generating one new chromosome (offspring) that initially contained a combination of the parents' characteristics. Since the population size was kept constant, whenever a new game was added to the current population, another game required removal. The applied replacement method was based on the Inverse Selection criterion (Rowe, 2015) and it was directly linked to the fitness function and the previously described selection method. This rule rendered the poorer performing solutions more likely to be replaced than the better ones, since the probability of being replaced was determined by the fitness level.

4.3.2 Reproduction: Crossover and Mutation

The reproduction method applied was a single-point crossover (Holland, 1992; Lucasius and Kateman, 1993), where one position was randomly chosen within the individual's binary structure and the parts of the two parents were exchanged at this point - generating a new individual, or offspring, as demonstrated in figure 7. The idea here was to recombine building blocks (schemes) on different strings. The crossover operator ensured that new individuals would inherit the characteristics of the parent chromosomes, which are likely to be high performers among the population, considering the probability of having higher payoff possibilities within their strategy vectors.

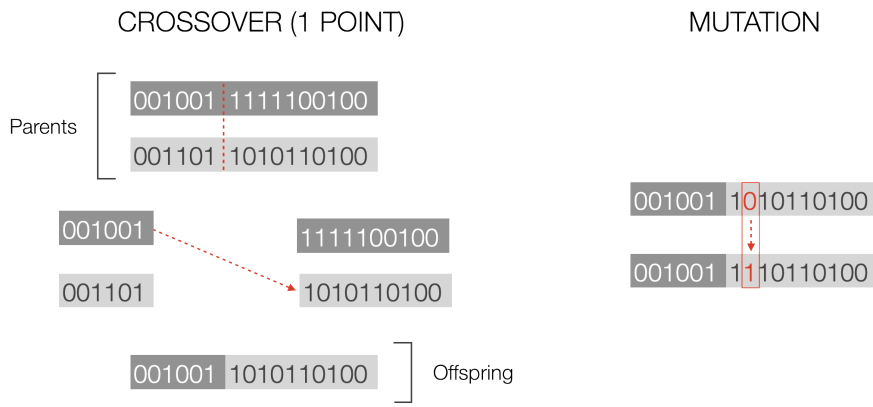

Figure 7: Crossover and Mutation Operators. Once the new individuals were created, the mutation operator insured the population against permanent fixation at any particular locus, thus playing a background role (Mitchell, 1998). In the application of mutation, one allele (or bit) of a gene was randomly replaced by another, and this factor created new games by making small and random changes in prior games (Eshelman et al., 2000). The standard rate of mutation applied was 1%. The reason for the adoption of this rate was due to further experimentation with differing probabilities which could have generated noisier results and lower optimization performance.

4.3.3 Evolutionary Loop

Once all the operations were performed for a determined generation, the final output (processed population) of a single run was re-inserted into the simulation as the input for the next run. The evolution loop was the master process that controlled all others and their data outputs by creating a genetic loop that would trigger the program's iteration until the stop criterion was satisfied, defined here as the evolution target. The populations of games and their fitness scores were the main object of analysis for accessing the algorithm properties and effects. The final data output consisted of a population of games with the same length as the initial set. This demonstrated on a global level the effects of N generations of simulated natural selection, breeding and mutation. Consequently, such generations directed the population to a convergence process towards an optimal point that was expected to become constant when first reached, meaning that the algorithm found an optimal solution for the problem provided. In summary, the steps applied were:

1. Initialized a pre-filtered population of 2×2 strategic form games

2. Implementation of the strategy selection methods outlined previously

3. Calculation of the fitness for each individual (game) in the population, based on the strategies performance (fitness proportionate selection).

4. Performed selection and genetic operators (crossover and mutation) within selected individuals in order to create a new game from the selected games.

5. Inserted the offspring back into the population, replacing a game by inverse selection.

6. Returned to step 2 and iterated this process until the termination criterion was satisfied

Parameter	**Value**	**Short Description**
Distinct populations	11	Populations restricted according to layers and families logic
Strategy selectors	3	Nash equilibrium, Hurwicz rule and Random
Selection rule	FPS	Fitness Proportionate Selection (roulette wheel)
Crossover points	1	Random point where the binary string is divided and recombined
Mutation rate	1%	Probability of mutation taking place
Replacement rate	1	Number of games replaced by the offspring in each generation
Termination	10000	Number of generations (iterations) allowed

Table 2: Summary: algorithmic parameters adopted in the model execution. The simulation model processed the populations and strategy selectors individually, iterating through the evolutionary loop until the termination criterion was satisfied.

5 Simulation Results

Overall, our simulation results indicated that the model mostly evolved in conformity with the presented literature and the expected development in accordance with the self-maximizing applied behavioural patterns. The evolutionary process drove the players into creating better outcomes throughout the learning generations, individually, by changing the structure of the games they were inserted into. The general trend was that when an optimal strategy was reached, the players had no incentives to deviate. Successful strategies were mostly repeated over time, creating an evolutionary equilibrium, as described in Riechmann (1998); Reschke et al. (2001); Riechmann (2002). When the equilibrium was disturbed by the stochastic mechanisms in place, the players tended to evolve again towards an optimal strategy. The mentioned conclusions hold true only for the rounds with the Nash and Hurwicz strategy selectors. The Random method generated fuzzier and unordered payoffs, mostly with non-optimal results.

We started with a game pool containing the following equilibrium structure: one pure Nash equilibrium (75% of the games), one completely mixed Nash equilibrium (12.5% of the games) and two pure and mixed (one or infinite many) Nash equilibrium (12.5% of the games). Table 3, depicts the resulting equilibrium structure, depending on the strategy selection method, of the new games generated by the evolutionary process, considering the games in the last period of the simulation rounds.

Equilibrium Structure	**Initial Pools**	**Hurwicz Pools**	**Nash Pools**	**Random Pools**
1 pure	**75.0%**	74.8%	48.7%	0.0%
1 completely mixed	**12.5%**	0.0%	0.0%	0.0%
2 pure (and mixed)	**12.5%**	25.2%	51.3%	100.0%
Total	**100.0%**	100.0%	100.0%	100.0%

Table 3: Relative frequencies of the equilibrium structure for games in the final populations, compared to the initial pools. Games with completely mixed Nash equilibria have been eliminated during the evolutionary process, making all new games with at least one pure Nash equilibrium. The symmetry in the payoff structures were not always present. A highlighted finding was the existence of an optimal equilibrium cell $(4, 4)$, where both players attained the maximum payoffs by playing that strategy, as in the Win-Win game family.

The most prominent findings emerging from the simulation data analyses, revealed four interesting points: the convergence to the optimal utility levels for both game players; the distinctive performance of the applied strategy selectors; the evolutionary stability of the adopted strategies; how the games transformed in the evolutionary process. Each of these points are explored more in depth hereunder.

5.1 Utility Convergence

Analysis of the utility development was plotted in figures 8, 9 and 10 (using average figures) for each of the populations and strategy selectors, split into two combined plots separately displaying the utility development for both players, ROW and COL. One can notice that in most cases, the charts displayed similar curves. The evolutionary process gradually increased the individual utility of the agents until the optimal value was reached. Consequently, the utility rates were rather constant. The agents proceeded to select strategies that maximized the individual payoffs as a means for maximizing the overall outcomes of the games themselves, as defined by the fitness function. In this manner, when all games in the given population achieved optimal payoffs, the equilibrium became stable, where the selected strategies tended to persist and resist other invading strategies (Friedman, 1991).

It is important to notice, however, that due to the stochastic mechanisms in the process generated by the Genetic Algorithm, some games in the population could have eventually lost an optimal pair of strategies during the evolutionary process. This is the reason for the noise in the utility curves lines the optimum was reached. Nevertheless, the subsequent generations quickly adapted to the optimal strategies again. Similar effects of genetic experiments in the economics field by Riechmann (1998, 2001, 2002), affirming that the learning process is given in two states: (1) the movement of populations towards a stable state, denominated behavioural stability and (2), once such state has been reached, the learning process presents a near-equilibrium dynamic of getting out of the evolutionarily stable state and returning there again, in conformity with the utility development.

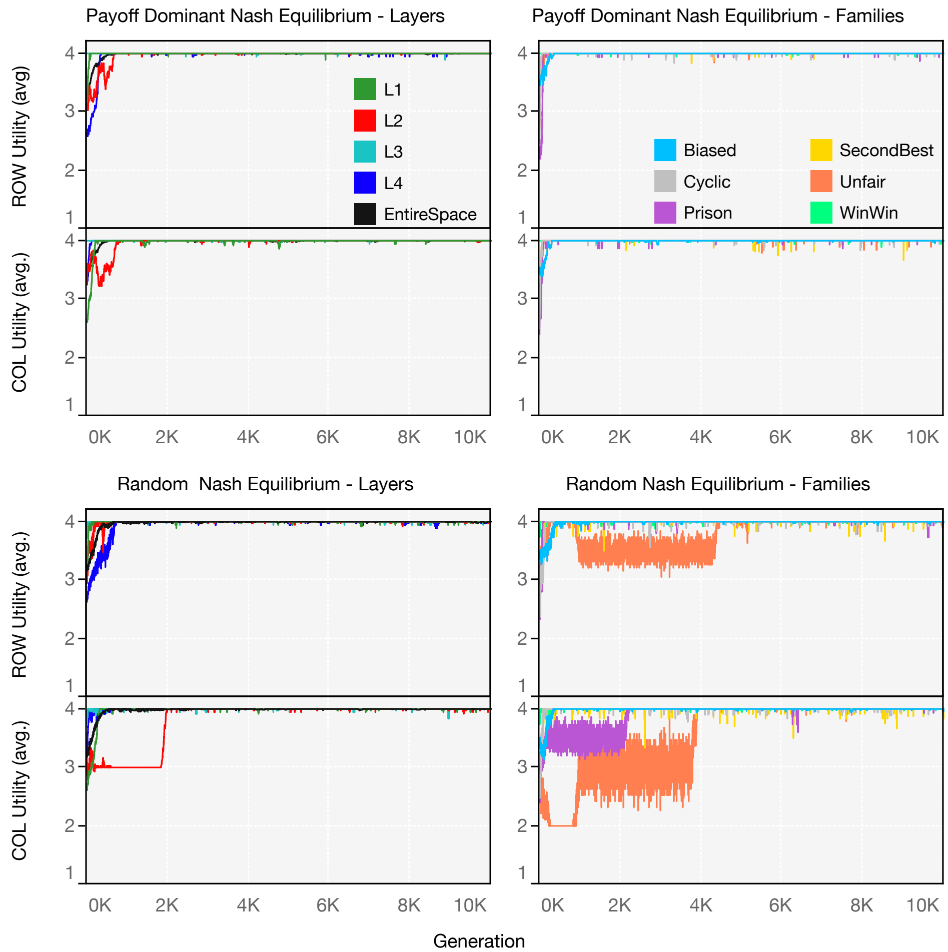

Figure 8: Utility Development - Nash equilibrium Strategies. The payoff dominant version of the Nash equilibrium strategy selector allowed the population to reach the optimal payoffs within a few hundred generations. Interestingly, the population that took the longer to achieve the equilibrium was *Layer 2* (left side), which contained the most *Biased* games, mixed with *Unfair*, *Second Best* and *Prison* games. Similarly, when compared to the population containing all *Biased* games (right side), they also displayed a lower convergence time, even when compared with games having initially inferior strategies. Analysis inferences from the random equilibrium version of the Nash equilibrium strategy selection, apart from the inferior performance to the payoff maximizing version (as expected), include the fact that the more diverse pools (layers division) presented a better performance than the pools with all similar games together (families division). In this case, the pools with *Biased*, *Unfair* and *Prison* games displayed a significantly higher convergence time.

Following data analyses of the populations using the Hurwicz rule, the equilibrium state was reached at similar speeds (see Figure 9) when compared to the Nash equilibrium runs.

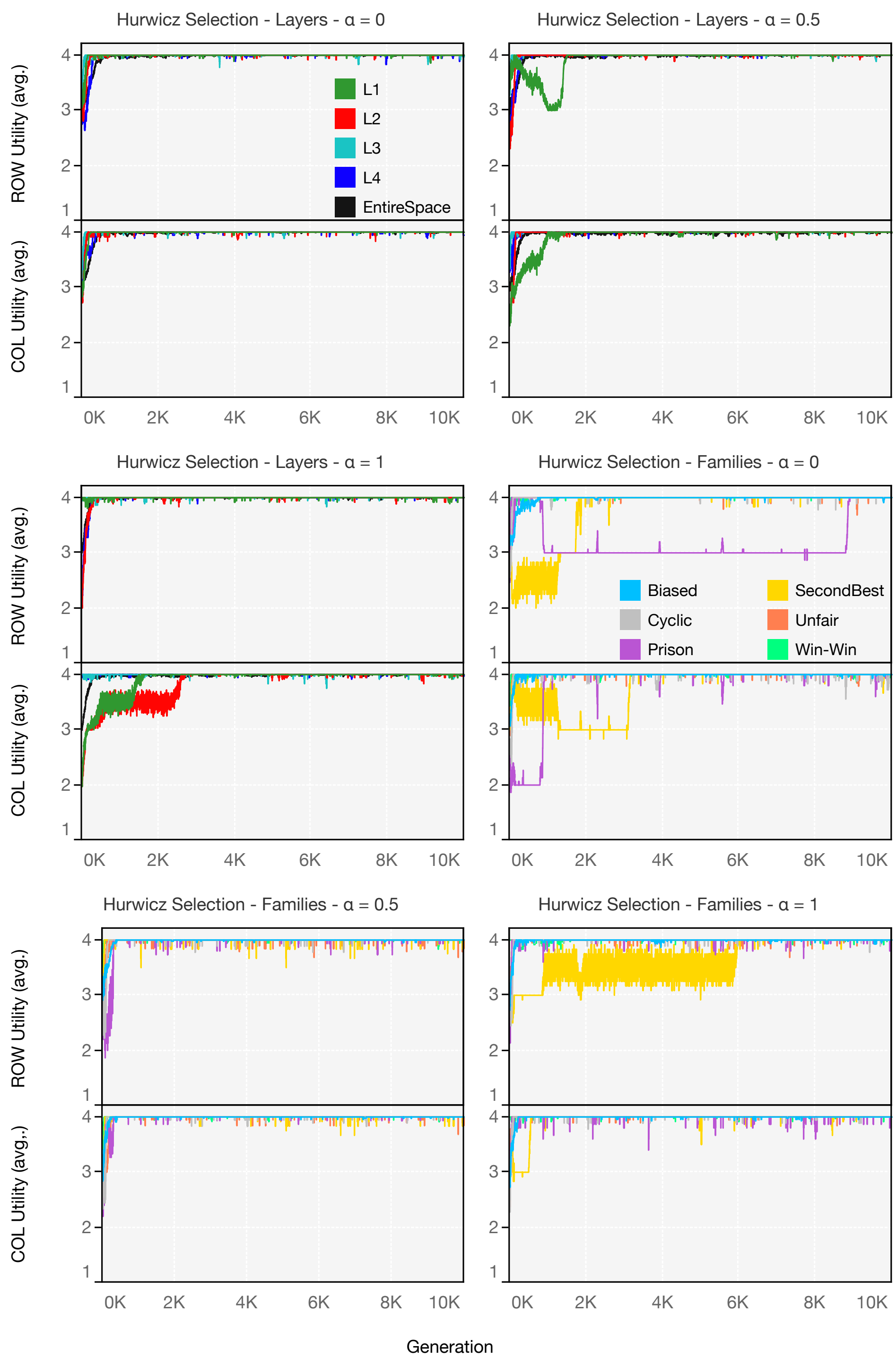

Figure 9: Utility Development - Hurwicz rule Strategies. According to the simulation results, the layers-based families presented similar effects as before, with *Layers 1* and *2* being slower than the others in some cases. In the families-based populations, there was a more distinct evolutionary pattern until stability was reached. The *Second Best* and *Prison* families demonstrated a much noisier and longer process to reach the equilibrium state, when considering the optimistic and, even more so, the pessimistic approaches

The random pools displayed an interesting control feature, exhibiting a noisier process and with marked lower populations that couldn't converge to the evolutionary stability, even after 10000 iterations, for both layers- and families-based populations (figure 10). This is an important contrast to the other runs, implying that the evolutionary process itself, even if tailored to maximize the payoffs, was not sufficient to drive the equilibrium state. The simulation of decision behaviour, in the context of this study, demonstrated key importance in driving the evolutionary learning process.

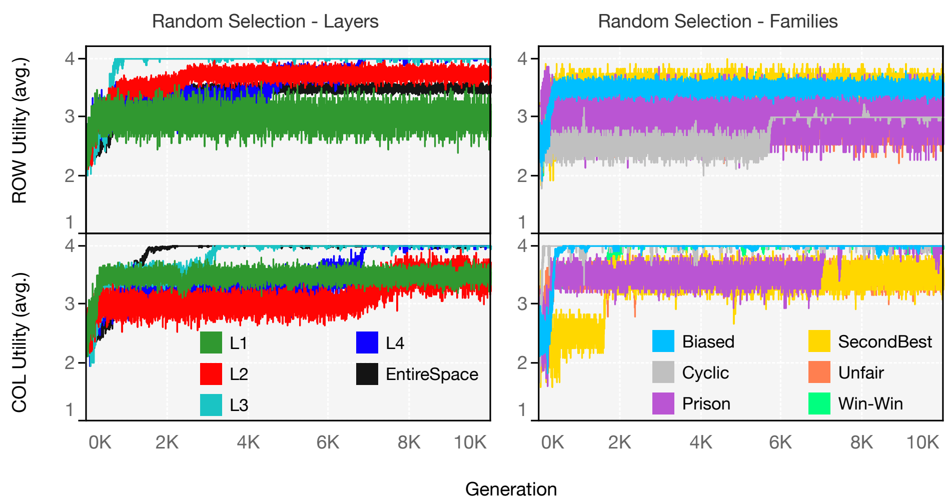

Figure 10: Utility Development - Random Strategies. This feature exhibited a noisier process and with most of the populations failing to maximize the players' utilities in a way that would allow for a stable equilibrium, even after 10000 iterations, for both layers- and families-based populations.

5.2 Strategies Performance

This section analyses the isolated strategy selectors' performance, aggregated by the average utility figures across all populations, in terms of utility yielded throughout the evolutionary process in each generation, together with analyzing the speed of convergence. The charts in figure 11 demonstrate the utility development for both players, by adopting each of the strategy selectors and variations used in the simulation model.

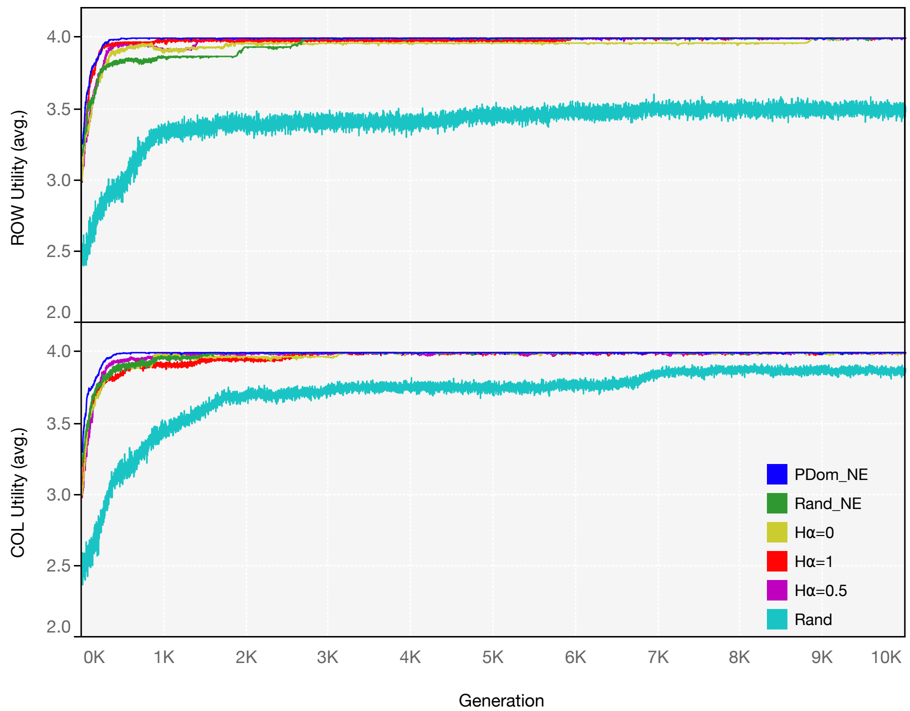

Figure 11: Strategies Performance - Consolidated. The payoff-dominant Nash equilibrium variant presented the highest performance when compared to the other models, for both players. The other strategy selectors presented performance variations for each of the players, though eventually reached equilibrium. An exception were the random strategy selectors, where for both players it frequently failed to lead the converge to the optimal utility figures. The performance of the strategies varied on an individual basis; a complete overview of each combination of game pool and strategy selector is presented in appendix A.

5.3 Evolutionary Stability - Coordination

The overall demonstration of the frequency of selected strategies by each player is displayed in figure 12, which contains a sample of randomly selected combinations of populations and strategy selectors. Following our 2×2 games structure, each player could select between two strategies, denominated as $Strategy1$ and $Strategy2$. Additionally, for the Nash equilibrium-based pools, there was the option of mixed strategies. One could notice that the evolutionary stability, in terms of utility reflected directly in the stability of the strategies adopted by each player, making the choices stable across the process. This trend was in line with the examples detailed in Weibull (1997); Hines (1987). Moreover, the majority displayed a symmetric plot, especially when the equilibrium was reached.

The literature suggests that in equilibrium situations, the players co-ordinate their strategies, meaning that for each strategy one of the players adopts, there is a strategy that is always the best response, giving no incentives for the players to deviate from this equilibrium state by selecting another strategy. This fact allows the equilibrium state to persist throughout the generations (see Figure 12).

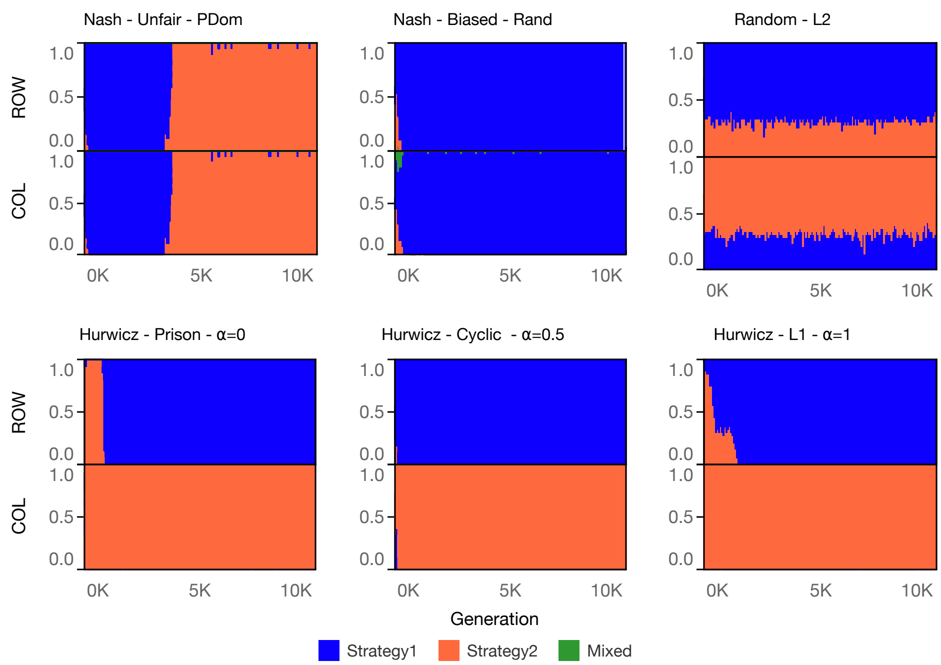

Figure 12: Evolutionary Stability - Sample Pools. Even with the changes in the utility figures proportioned by stochastic effects on the structure of games, in most cases (mostly all, except for the pools where the strategies were selected randomly), the players did not deviate from well-performing strategies, keeping the decision-making behaviour evolutionarily stable against invading strategies (Weibull, 1997). It could have been the case, however, that occasionally the strategies changed for one or for both players accordingly, as changes in the structure of the games could have shifted the equilibrium state and forced players to change their strategy in order to keep the optimal utility. This figure displays a random sample of six combinations of strategy selectors and game pools. The complete overview of the frequency of the selected strategies in all pools is found in appendix B.

5.4 Transformation of Games

Another interesting fact observed during the analysis of the simulation results is how the games transformed and replicated, reducing the number of unique games in the population as the evolutionary process progressed. Successful games were replicated and appeared repeatedly in the final populations. There was no restriction to the manners in which the game could change, except for the rules defining the periodic table of 2×2 games (Robinson and Goforth, 2005). Even in cases when the payoff symmetry was broken, there was an optimal equilibrium state for both players which tended to survive across generations. The games were changed in such a way that it retained (in the majority of cases) the strategy choices and payoffs yield stabilised, according to the characteristics of the initial populations and the decision rules (strategy selectors) applied. In this case, the games were specifically optimized to create favourable decision scenarios for both players.

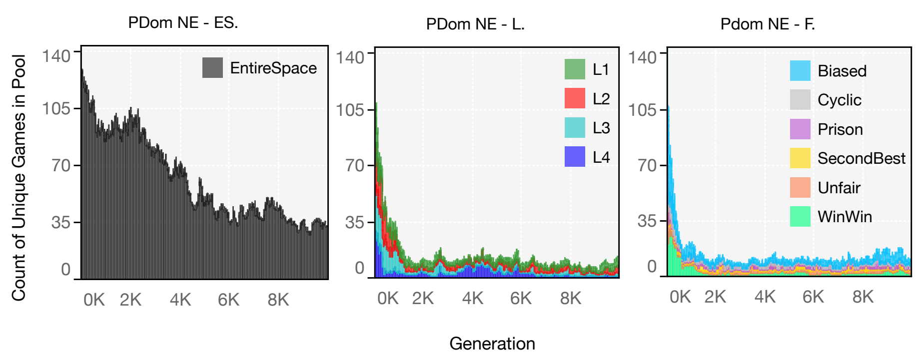

Figure 13: Count of Unique Games in Pool Across Generations - Sample Pools. All cases demonstrated the same decreasing trends. This figures displays a random sample of three combinations of strategy selectors and game pools. The complete overview of the count of games in all pools is found in appendix C.

This fact is demonstrated in figure 13, which contains a random sample of pools plotting the count of unique games in each of the denoted populations. This reduction pattern was equivalent across every population and strategy selector. The model yielded populations with a few different games, occurring repeatedly within the same pool.

6 Discussion

The simulation model documented in this article aimed to deliver an experimental and exploratory approach, focusing on the analysis of the impact of individual decision behaviour on strategic interaction scenarios. Based on how economic agents learn and transform their decision environments, the model introduced here was built using Genetic Algorithms and game theory, specifically consisting of the 2×2 strategic form games, using the representation provided in Robinson and Goforth (2005); Bruns (2010, 2011, 2015a,b) for simulating diverse strategic scenarios. Through playing the same games with different decision rules (the Nash equilibrium, the Hurwicz rule and a random choice method), one can compare the effects of different decision behaviours in similar games and how such games evolve over time, when favouring self-maximizing behaviour profiles and concomitantly punishing poor performing strategies, relative to the results of its peers.

The core observations extracted from the presented results can be summarized in the following points:

- When agents maximize their own utility, the games evolve to have higher payoff structures, optimizing their own decision environments.
- Over time, conflict situations transformed into "Win-Win" scenarios that favoured both players, based on their simulated behavioural aspects.
- The evolution of dynamic strategic situations tends to create games where co-ordination is possible.
- Decision rules and the environment characteristics influence the agility of the learning process, in terms of reaching an evolutionarily stable equilibrium.

The diverse set of games used for this analysis provided a rich representation of multiple real-life scenarios, including conflict, biased, dilemma, and favourable situations. In addition, the combination of multiple game types with multiple behaviour types yielded a diverse data set for exploring additional individual factors, such as which strategies perform better, and which types of games are the most challenging to achieve a stable equilibrium.

The introduced decision rules, as the agents' strategy selectors, presented a satisfactory performance in supporting the transformation of the games with the objective of maximizing individual payoffs. An exception was the random selection method, which failed to reach the evolutionary equilibrium multiple times in the allowed time. In many points, the performance of the Nash equilibrium and Hurwicz rule variations are similarly ranked. However, in essence, the Nash equilibrium was still found to be, in this study, the most rapid and robust method for optimization modelling, in conformity with past findings by Sefrioui and Perlaux (2000).

The self-maximizing behaviour and the transformation of the games with the Genetic Algorithms allowed the individual strategies to evolve in such a way that the novel and modified games allow the players to co-ordinate their strategies. This means that for each of one player's strategies, there is an optimal best response. This transmutation of the games allows both players to maximize their own payoffs in a stable manner across generations. When the genetic process eliminates stable strategies, the evolutionary learning process eventually creates new optimal strategies that enable a new equilibrium state, reinforcing the findings in Kalai and Lehrer (1993); Chmura et al. (2012), that rational learning in repeated games eventually leads to stationary points of the Nash equilibrium. This finding is also in line with the concept of genetic stability of equilibrium populations in economic applications of Genetic Algorithms, documented in Riechmann (1998, 2001, 2002). Over time, repeated strategic interactions represented as games, are influenced by the players and consequently transformed into favourable scenarios in all cases. This means that self-maximizing computational agents can learn from their actions and transform conflict situations over time, as they learn how to behave optimally, given the structural information and constraints they are exposed to.

It is clear, however, that evolutionary social models - in reality - are more complex and require high efforts in order to create representative pictures that capture relevant characteristics of social systems (Reschke et al., 2001). As discussed in Erev and Roth (1998); Brown and Rosenthal (1990); Chmura et al. (2012), actual human behavior is at times not well predicted by the standard theory. The model created in this study aimed at an exploratory analysis that contributes to the discussion on how to envision and build normative and representative models in many different situations, to enable humans to make better decisions and allow the exploration of different scenarios and time factors in a reasonable and efficient way.

Further research can be performed by enriching the current model with different types of games, as well as encoding different strategy selection models which should capture more diverse behavioural profiles of decision makers in varying contexts. Furthermore, practitioners can add another layer of processing payoffs and utilities by defining a spectrum of profiles based on social preferences, such as altruism, envy, fairness and justice, defined by the agents' utility functions. These changes can introduce more behavioural complexity in the analysis (such as the overview provided in Angner (2012), chapter 11). Such enhancements would directly affect the evolutionary process by altering the fitness function through individual payoffs.

References

Adami, C., Schossau, J., and Hintze, A. (2016). Evolutionary game theory using agent-based methods. *Physics of life reviews*, 19:1–26.

Angner, E. (2012). *A course in behavioral economics*. Macmillan International Higher Education.

Axelrod, R. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration*, volume 3. Princeton University Press.

Axelrod, R. et al. (1987). The evolution of strategies in the iterated prisoner's dilemma. *The dynamics of norms*, pages 1–16.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Baddeley, M. (2018). *Behavioural economics and finance*. Routledge.

Billari, F. C., Fent, T., Prskawetz, A., and Scheffran, J. (2006). Agent-based computational modelling: an introduction. In *Agent-Based Computational Modelling*, pages 1–16. Springer.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3):7280–7287.

Brown, J. N. and Rosenthal, R. W. (1990). Testing the minimax hypothesis: a re-examination of o'neill's game experiment. *Econometrica: Journal of the Econometric Society*, pages 1065–1081.

Brunner, C., Camerer, C., and Goeree, J. K. (2010). A correction and re-examination of'stationary concepts for experimental 2x2 games'. *American Economic Review, forthcoming.*

Bruns, B. (2010). Navigating the topology of 2x2 games: an introductory note on payoff families, normalization, and natural order. *arXiv preprint arXiv:1010.4727*.

Bruns, B. (2011). Visualizing the topology of 2x2 games: From prisoner's dilemma to win-win. *In Stony Brook, NY: Game Theory Center*.

Bruns, B. (2015a). Atlas of 2x2 games: Transforming conflict and cooperation.

Bruns, B. R. (2015b). Names for games: locating 2× 2 games. *Games*, 6(4):495–520.

Camerer, C. and Hua Ho, T. (1999). Experience-weighted attraction learning in normal form games. *Econometrica*, 67(4):827–874.

Camerer, C. and Weigelt, K. (1988). Experimental tests of a sequential equilibrium reputation model. *Econometrica: Journal of the Econometric Society*, pages 1–36.

Camerer, C. F. (2003). Strategizing in the brain. *Science*, 300(5626):1673–1675.

Camerer, C. F. (2011). *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press.

Castillo, L. and Dorao, C. (2013). Decision-making in the oil and gas projects based on game theory: Conceptual process design. *Energy Conversion and Management*, 66:48–55.

Castillo, L. and Dorao, C. A. (2012). Consensual decision-making model based on game theory for lng processes. *Energy conversion and management*, 64:387–396.

Cheung, Y.-W. and Friedman, D. (1997). Individual learning in normal form games: Some laboratory results. *Games and economic behavior*, 19(1):46–76.

Chmura, T., Goerg, S. J., and Selten, R. (2012). Learning in experimental 2×2 games. *Games and Economic Behavior*, 76(1):44 – 73.

Chmura, T., Kaiser, J., and Pitz, T. (2007). Simulating complex social behaviour with the genetic action tree kernel. *Computational and Mathematical Organization Theory*, 13(4):355–377.

Colman, A. M. (2016). *Game theory and experimental games: The study of strategic interaction*. Elsevier.

Dhami, S. (2016). *The foundations of behavioral economic analysis*. Oxford University Press.

Erev, I. and Roth, A. E. (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American economic review*, pages 848–881.

Eshelman, L. J., Bäck, T., Fogel, D., and Michalewicz, T. (2000). Genetic algorithms. *Evolutionary Computation*, 1:64–80.

Freedman, L. (2017). *The transformation of strategic affairs*. Number 379. Routledge.

Friedman, D. (1991). Evolutionary games in economics. *Econometrica: Journal of the Econometric Society*, pages 637–666.

Fudenberg, D. and Tirole, J. (1991). *Game theory*. MIT press.

Gaspars-Wieloch, H. (2014). Modifications of the hurwicz's decision rule. *Central European*

Journal of Operations Research, 22(4):779–794.
Gintis, H. et al. (2000). *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton university press.
Glynatsi, N. E., Knight, V., and Lee, T. E. (2018). An evolutionary game theoretic model of rhino horn devaluation. *Ecological Modelling*, 389:33–40.
Goldberg, D. E. and Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99.
Goldman, A. I. (1971). The individuation of action. *The journal of Philosophy*, 68(21):761–774.
Gooding, T. (2014). Modelling society's evolutionary forces. *Journal of Artificial Societies and Social Simulation*, 17(3):3.
Hamblin, S. (2013). On the practical usage of genetic algorithms in ecology and evolution. *Methods in Ecology and Evolution*, 4(2):184–194.
Hamblin, S. and Hurd, P. L. (2007). Genetic algorithms and non-ess solutions to game theory models. *Animal Behaviour*, 74(4):1005–1018.
Harley, C. B. (1981). Learning the evolutionarily stable strategy. *Journal of theoretical biology*, 89(4):611–633.
Harsanyi, J. C., Selten, R., et al. (1988). A general theory of equilibrium selection in games. *MIT Press Books*, 1.
Haupt, R. L., Haupt, S. E., and Haupt, S. E. (1998). *Practical genetic algorithms*, volume 2. Wiley New York.
Hayashi, N., Ostrom, E., Walker, J., and Yamagishi, T. (1999). Reciprocity, trust, and the sense of control: A cross-societal study. *Rationality and society*, 11(1):27–46.
Heckathorn, D. D. (1996). The dynamics and dilemmas of collective action. *American sociological review*, pages 250–277.
Hines, W. (1987). Evolutionary stable strategies: a review of basic theory. *Theoretical Population Biology*, 31(2):195–272.
Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–73.
Hurwicz, L. (1951). The generalized bayes minimax principle: a criterion for decision making under uncertainty. *Cowles Comm. Discuss. Paper Stat*, 335:1950.
Isaac, A. G. (2008). Simulating evolutionary games: a python-based introduction. *Journal of Artificial Societies and Social Simulation*, 11(3):8.
Jaffray, J.-Y., Jeleva, M., Gains, U., and Paris, C. (2007). Information processing under imprecise risk with the hurwicz criterion. In *5th international symposium on imprecise probability: theories and applications*, pages 233–242. Citeseer.
Kalai, E. and Lehrer, E. (1993). Rational learning leads to nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 1019–1045.
Knight, V. and Campbell, J. (2018). Nashpy: A python library for the computation of nash equilibria. *Journal of Open Source Software*, 3(30):904.
Kollock, P. (1998). Social dilemmas: The anatomy of cooperation. *Annual review of sociology*, 24(1):183–214.
Kramer, O. (2017). Genetic algorithm essentials: Studies in computational intelligence.
Kusyk, J., Sahin, C. S., Uyar, M. U., Urrea, E., and Gundry, S. (2011). Self-organization of nodes in mobile ad hoc networks using evolutionary games and genetic algorithms. *Journal of Advanced Research*, 2(3):253–264.

Liu, Y., Tang, W., He, J., Liu, Y., Ai, T., and Liu, D. (2015). A land-use spatial optimization model based on genetic optimization and game theory. *Computers, Environment and Urban Systems*, 49:1–14.

Lucasius, C. B. and Kateman, G. (1993). Understanding and using genetic algorithms part 1. concepts, properties and context. *Chemometrics and intelligent laboratory systems*, 19(1):1–33.

Maschler, M., Solan, E., and Zamir, S. (2013). *Game Theory*. Cambridge University Press.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

Mohamed, F. A. and Koivo, H. N. (2011). Multiobjective optimization using modified game theory for online management of microgrid. *European Transactions on Electrical Power*, 21(1):839–854.

Morgenstern, O. and Von Neumann, J. (1953). *Theory of games and economic behavior*. Princeton university press.

Pangallo, M., Sanders, J., Galla, T., and Farmer, J. D. (2020). A taxonomy of learning dynamics in 2× 2 games. *A Taxonomy of Learning Dynamics in*, 2(2).

Pavlov, P. I. (2010). Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex. *Annals of neurosciences*, 17(3):136.

Pažek, K. and Rozman, Č. (2009). Decision making under conditions of uncertainty in agriculture: a case study of oil crops. *Poljoprivreda*, 15(1):45–50.

Pereira, R. L., Souza, D. L., Mollinetti, M. A. F., Neto, M. T. S., Yasojima, E. K. K., Teixeira, O. N., and De Oliveira, R. C. L. (2020). Game theory and social interaction for selection and crossover pressure control in genetic algorithms: An empirical analysis to real-valued constrained optimization. *IEEE Access*, 8:144839–144865.

Périaux, J., Chen, H., Mantel, B., Sefrioui, M., and Sui, H. (2001). Combining game theory and genetic algorithms with application to ddm-nozzle optimization problems. *Finite elements in analysis and design*, 37(5):417–429.

Pitz, T., Chmura, T., et al. (2005). Genetic action trees a new concept for social and economic simulation. Technical report, University Library of Munich, Germany.

Puerto, J., Mármol, A., Monroy, L., and Fernández, F. (2000). Decision criteria with partial information. *International Transactions in Operational Research*, 7(1):51–65.

Reeves, C. and Rowe, J. E. (2002). *Genetic algorithms: principles and perspectives: a guide to GA theory*, volume 20. Springer Science & Business Media.

Reschke, C. H. et al. (2001). Evolutionary perspectives on simulations of social systems. *Journal of Artificial Societies and Social Simulation*, 4(4):10.

Riechmann, T. (1998). Genetic algorithms and economic evolution. Technical report, Diskussionsbeitrag.

Riechmann, T. (2001). Genetic algorithm learning and evolutionary games. *Journal of Economic Dynamics and Control*, 25(6-7):1019–1037.

Riechmann, T. (2002). Genetic algorithm learning and economic evolution. In *Evolutionary computation in Economics and Finance*, pages 45–60. Springer.

Robinson, D. and Goforth, D. (2005). *The topology of the 2x2 games: a new periodic table*, volume 3. Psychology Press.

Roca, C. P., Cuesta, J. A., and Sánchez, A. (2009). Evolutionary game theory: Temporal and spatial effects beyond replicator dynamics. *Physics of life reviews*, 6(4):208–249.

Rowe, J. E. (2015). Genetic algorithms. In Kacprzyk, J. and Pedrycz, W., editors, *Springer*

Handbook of Computational Intelligence, pages 825–844. Springer, Berlin, Heidelberg.

Samuelson, L. (1997). *Evolutionary games and equilibrium selection*, volume 1. MIT press.

Sefrioui, M. and Perlaux, J. (2000). Nash genetic algorithms: examples and applications. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, volume 1, pages 509–516. IEEE.

Simpson, B. (2004). Social values, subjective transformations, and cooperation in social dilemmas. *Social Psychology Quarterly*, 67(4):385–395.

Tanimoto, J. (2015). *Fundamentals of evolutionary game theory and its applications*. Springer.

von Stengel, B. (2007). Equilibrium computation for two-player games in strategic and extensive form. *Algorithmic game theory*, pages 53–78.

Weibull, J. W. (1997). *Evolutionary game theory*. MIT press.

Widger, J. and Grosu, D. (2008). Computing equilibria in bimatrix games by parallel support enumeration. In *2008 International Symposium on Parallel and Distributed Computing*, pages 250–256. IEEE.

Yang, G. (2017). Game theory-inspired evolutionary algorithm for global optimization. *Algorithms*, 10(4):111.

7 Appendix A: Complete diagram of strategies performance

Figure 14 contains the overview of the performance achieved by the different decision rules in every population, individually

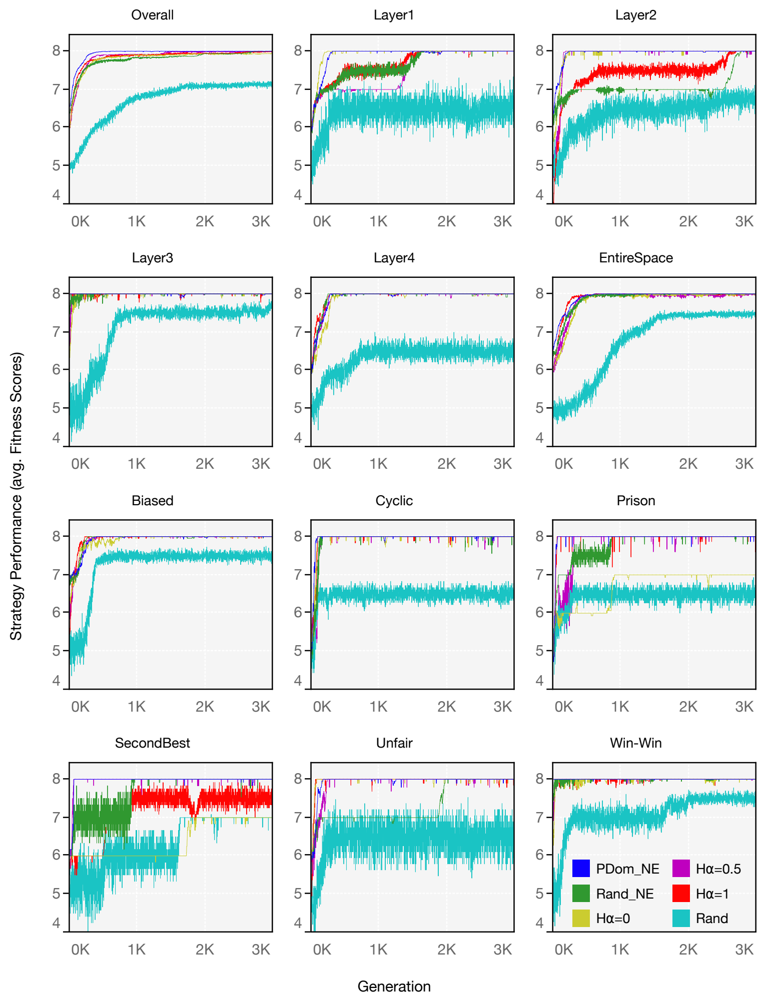

Figure 14: Strategies Performance - All Pools

8 Appendix B: Complete diagram of evolutionarily stable strategies

Complete overview of the strategies selection frequency for the Nash equilibrium Pools in figure 15

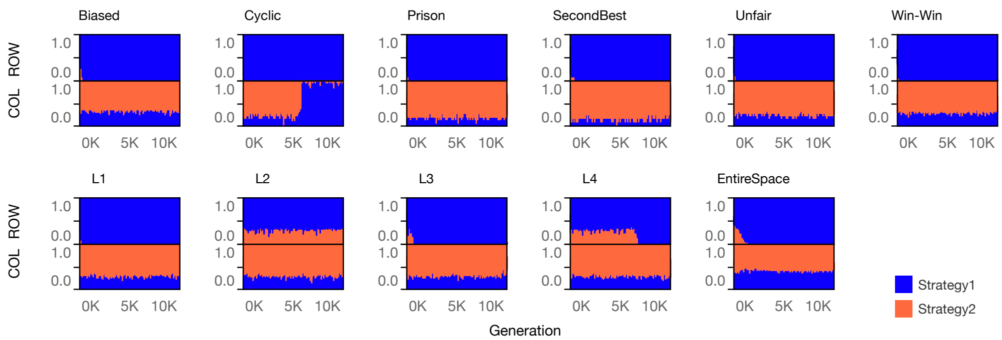

Figure 15: Evolutionary Stability - Nash Pools

Complete overview of the strategies selection frequency for the Hurwicz Pools in figure 16

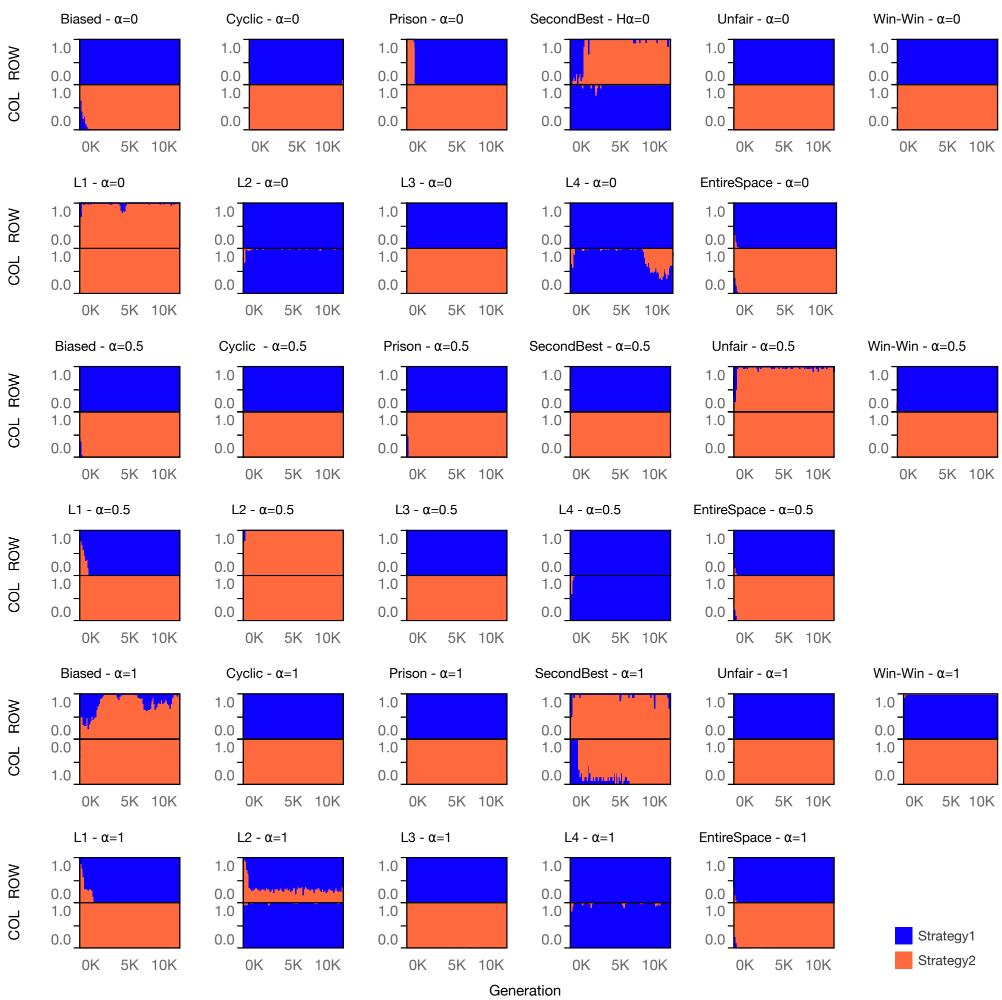

Figure 16: Evolutionary Stability - Hurwicz Pools

Complete overview of the strategies selection frequency for the Random Pools in figure 17

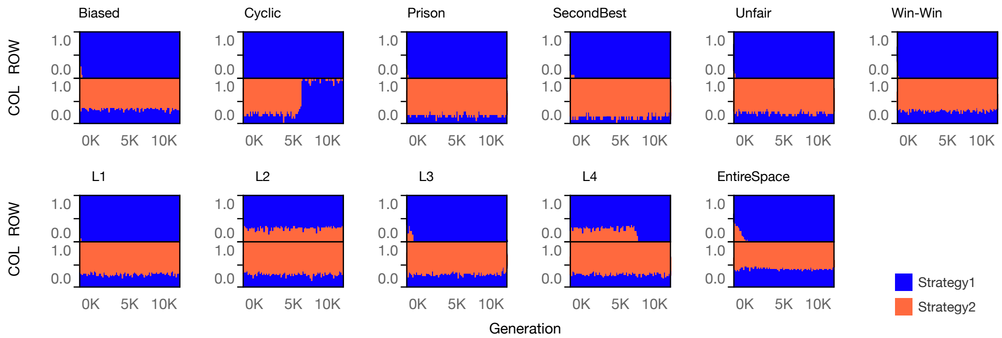

Figure 17: Evolutionary Stability - Random Pools

9 Appendix C: Complete diagram of games transformation

Figure 18 displays the overview of the count of unique games across generations in all pools, for all strategy selectors.

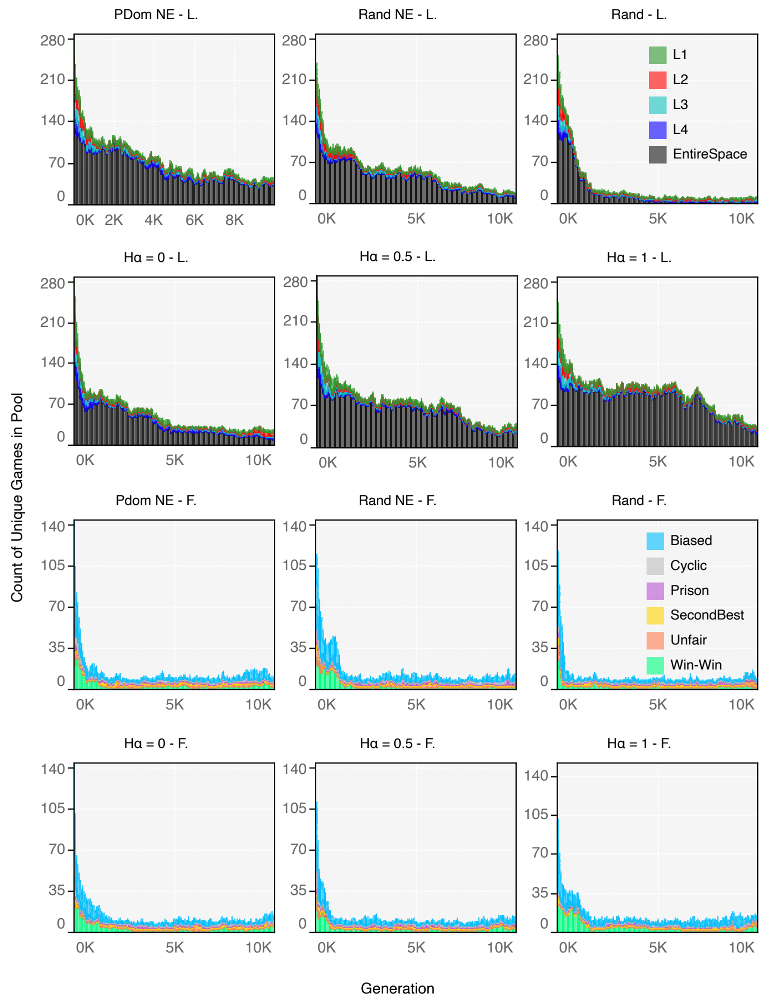

Figure 18: Count of Unique Games in Pool Across Generations - All Pools