



Analyzing the Impact of Strategic Behavior in an Evolutionary Learning Model Using a Genetic Algorithm

Vinícius Ferraz¹ · Thomas Pitz²

Accepted: 17 November 2022 / Published online: 14 December 2022
© The Author(s) 2022

Abstract

This study presents an experimental approach to strategic behavior and economic learning by integrating game theory and Genetic Algorithms in a novel heuristic-based simulation model. Inspired by strategic scenarios that change over time, we propose a model where games can change based on agents' behavior. The goal is to document the model design and examine how strategic behavior impacts the evolution of optimal outcomes in various choice scenarios. For diversity, 144 unique 2×2 games and three different strategy selection criteria were used: Nash equilibrium, Hurwicz rule, and a random selection technique. The originality of this study is that the introduced evolutionary algorithm changes the games based on their overall outcome rather than changing the strategies or player-specific traits. The results indicated optimal player scenarios for both The Nash equilibrium and Hurwicz rules, the first being the best-performing strategy. The random selection method failed to converge to optimal values in most of the selected populations, acting as a control feature and reinforcing the need for strategic behavior in evolutionary learning. Two further observations were recorded. First, games were frequently transformed so agents could coordinate their strategies to create stable optimal equilibria. Second, we observed the evolution of game populations into groups of fewer (repeating) isomorphic games with strong preceding game characteristics.

Keywords Game theory · Simulation · Genetic algorithms · Economic learning · Artificial intelligence

JEL Classification C02 · C63 · C73

✉ Vinícius Ferraz
visferraz@gmail.com

Thomas Pitz
thomas.pitz@hochschule-rhein-waal.de

¹ Alfred-Weber Institute for Economics, University of Heidelberg, Heidelberg, Germany

² Faculty of Society and Economics, Rhine-Waal University of Applied Sciences, Kleve, Germany

1 Introduction

Our world is constantly going through systemic transformations. Technological and scientific advancements, in combination with changes in economic, political, sociological, and other factors, result in new decision contexts in which strategic interaction occurs. By analyzing such situations, one can observe the ever-arising need for individuals to update their beliefs and adapt to the new informational and strategic structures. As an example, Freedman (1998, 2017) reported the example of socioeconomic and political transformations occasioned by the development of technology and communication structures, resulting in changes in how nations engage in warfare conflict. When making decisions, economic agents will act according to their interests, as well as the action of other agents and information available in the given scenarios, as outlined by Von Neumann and Morgenstern (1953).

As stated by Axelrod (1997), one can understand the properties of complex social and economic systems by applying simulations. The nature of human interaction is often modeled and analyzed in computational models of society, which introduce autonomous agents that interact with one another and the environment into which they are placed according to predefined rules (Billari et al., 2006). Adding dynamics to models of strategic interaction, social learning, and the evolutionary process are often simulated by introducing evolutionary biology concepts, as outlined by Gintis (2000). This evolutionary approach introduces the notion of predefined strategies that are repeatedly applied in an evolutionary process, operating dynamically on the distribution of behavior (Weibull, 1997).

Game theory models describe strategic scenarios and behavior (Von Neumann & Morgenstern, 1953). In game theory, political or socioeconomic conflicts or crises are often modeled in a strategic form by matrix games or in an extensive form by game trees. One often restricts oneself to a fixed game that does not change significantly over time. Prominent examples are the analysis of the Cold War as a Prisoner Dilemma (Plous, 1993) or the Cuba crisis as a Chicken Game (Russell, 1959). In reality, however, it is observable that the strategic character of conflicts or crises changes over time. A crisis modeled as a Prisoner Dilemma can intensify into a Chicken Game with higher conflict potential or transform into a less conflictual Stag Hunt (Skyrms, 2004) or Harmony Game (Bruno, 2010). Therefore, it would be appropriate to describe these strategic changes by transforming the original game into a new game. In empirical settings, fundamental behavioral changes are observed when making decisions that can affect not only the strategic behavior of the agents involved but also environmental conditions and individual preferences. Heckathorn (1996) documented the transformation of games with dynamic interaction, where changes in decision-influencing factors changed the whole structure of the initial game. Similarly, Simpson (2004) empirically demonstrated how behavioral factors, such as social preferences, can transform one game into another (see also Hayashi et al. (1999); Kolllock (1998)).

Motivated by this fact, in the present paper, we introduce a novel heuristic procedure that describes these changes in strategic interaction scenarios with Genetic Algorithms, a search procedure inspired by the process of biological evolution (Holland, 1992; Goldberg & Holland, 1988). Genetic Algorithms have been employed in economics-based problems since their introduction and are regarded as a powerful tool for finding optimal solutions over complex search spaces (Schmertmann, 1996), as well as a method that shows remarkable results for the simulation of decision behavior that is in line with empirical observations in similar research frameworks (Arifovic & Ledyard, 2012; Lensberg & Schenk-Hoppé, 2021; Manson, 2006). Given a pool of 2×2 strategic-form games, the games, represented as binary sequences, are transformed by a Genetic Algorithm depending on the players' experience with the games in the pool. In each round, after making decisions, the game is evaluated based on the players' payoffs. This process determines the probability of staying in the pool, or being replaced by another new game created via crossover and mutation. We describe the dynamics of the game pools related to the different types of strategy selection rules adopted by the players, establishing the notion of strategic behavior for the agents. The populations are defined based on two rules for aggregating sets of games. The first rule is based on topological proximity, using the periodic table of 2×2 games concept introduced by (Robinson & Goforth, 2005); the second rule clusters games by similar characteristics, based on the families categorization introduced in Bruns (2010, 2015a).

The introduced model focuses on analyzing the impact of strategic behavior in this evolutionary learning process, where games are allowed to change over time and the performance of different decision rules. We aimed to document the implementation, testing, and assessing different decision-making rules and their influence on dynamic game populations. For a comprehensive analysis, the simulation model described here analyzes 144 unique types of 2×2 games and three distinct strategy selection rules: Nash equilibrium, Hurwicz rule, and a Random selection method. The goal is to outline how strategic behavior affects the transformation of dynamic decision-making scenarios by pairing different strategy selection rules with distinct populations of 2×2 games. The analysis of the simulation results focused on convergence speed (optimal utility levels reached) using different combinations as a performance measure, as the encoding of strategies and convergence process are seemingly interconnected (Dawid & Kopel, 1998). We have also documented the findings derived from the simulation process, including the consistent transformation of games and behavioral traits highlighted during the process.¹

¹ The documented source code, data, and resources used in this paper are published in the CoMSES Library. The material is available to download at: <https://doi.org/10.25937/smg0-0t92>.

2 Evolutionary Computation Overview

Evolutionary computation methods arose from taking inspiration from biological mechanisms to design and implement computer-based problem-solving systems (Spears et al., 1993). This collection of methods allowed the creation of evolving and adaptive solutions to complex problems, especially the ones that impose challenges to traditional algorithms, such as randomness, chaotic disturbances, and complex non-linear dynamics, as outlined by Fogel (2000). The family of evolutionary algorithms contains several different methods, each with its particularities, but all of them share a connection with biological evolution. Among the most known methods in the literature are Genetic Algorithms (Holland, 1992), Genetic Programming (Koza et al., 1992), Differential Evolution (Storn & Price, 1997), Evolution Strategies (Rechenberg, 1978) and Evolutionary Programming (Fogel, 1998). For an overview of each of these methods, see Slowik and Kwasnicka (2020). The literature discussion that follows next outlines applications of some of these methods in similar contexts, highlighting the elements that led to adopting a Genetic Algorithm as an appropriate method to simulate the dynamic transformations of games over time.

2.1 Evolutionary Models: Applications for Learning, Strategic Interaction, and Optimization

As Camerer (2003) outlined, some aspects of learning are sometimes overlooked by economic theory. If perfect information and rationality are assumed, the equilibrium point will always be known from the beginning, and people will only modify the equilibrium if information changes. Moreover, Camerer and Weigelt (1988) emphasized the importance of 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 crucial in providing predictive power, coherence, and concomitantly revealing new insights (Camerer, 2011).

The multidisciplinary combination of game theory and genetic programming has grown in several distinct fields, 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, in contrast with the neoclassic assumption of perfect rationality, economic models of learning provide the possibility to study agents as they learn and update their beliefs since the application of an evolutionary model assumes that strategies can change over time (Baddeley, 2018). According to Axelrod et al. (1987), individuals cannot thoroughly analyze the situation and calculate optimal strategies when interacting in complex environments. Alternatively, strategies are updated and based on achieved results, highlighting how a Genetic Algorithm can be particularly adept as a learning mechanism for creating effective strategies. The given approach serves as an inspiration for the analysis performed in this article. The following are some of the relevant constructs.

Holland and Miller (1991) stresses that the employment of artificial adaptive agents in economic theory can help us understand real-world economic issues by enabling the free exploration of system dynamics under controlled conditions and the opportunity to check several unfolding behavioral patterns. Furthermore, Dawid (1999) argued that the decentralized structure of Genetic Algorithms, which naturally resembles a group of interacting economic agents, is well-suited to simulate the behavior of economic systems.

Isaac (2008) provided an introductory overview of an agent-based model using Genetic Algorithms in the iterated Prisoner's Dilemma, reporting variations in the payoff structures that create 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 analyzing action patterns in social systems mainly based on the concepts of Genetic Algorithms and the Theory of Action Trees (Goldman, 1971). They explained how the emergence and disappearance of actions could be described with a uniform algorithm, succeeding in endogenously eliciting comprehensive changes in the agents' behavior. Manson (2006) documented experiments exploring the concept of bounded rationality, stating that Genetic Algorithms are an appropriate tool to model actors that are not perfectly rational, that is, addressing characteristics of human decision-making such as cognitive limits, learning, and innovation.

Similarly, the algorithm for optimization problems reported by Yang (2017) build on a similar conceptual framework. Their main idea is a game theory-inspired evolutionary model that updates the strategy sets by replacing individuals of the population with better-performing offspring generated by replication or belief learning operators, creating a model that outperforms four other algorithms 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 interactions (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 the engineering design optimization process. Continuing on the topic of optimization.

Savin and Egbetokun (2016) formulated an agent-based model of innovation networks with endogenic absorption capacity, where dynamic cooperation for knowledge can occur between different agents, represented as firms, with different knowledge positions. In their simulation model, the authors applied a Differential Evolution algorithm to find optimal investment budget decisions regarding trade-offs between cooperative and non-cooperative scenarios. Their findings demonstrate that networks generated with the model display small-world properties, which tend to be efficient structures for knowledge distribution. Another interesting observation is that firms with higher absorptive capacity tend to be better positioned within their networks, ultimately demonstrating that their ability to learn drives network performance effects.

With a similar objective to this paper, Savin et al. (2018) introduced a meta-heuristic approach for solving non-linear dynamic games, proposing a method that allows the analysis of more realistic strategic scenarios. The method can solve the

standard version of the introduced game, like other traditional techniques, and solve non-standard extensions of the problem (inequality constraint and asymmetry in penalties) by identifying optimal equilibrium strategies, both cooperative and non-cooperative. The proposed procedure combines Differential Evolution (for individual optimal strategies) and Approximation of a Nash Equilibrium. The set consists of a three-player macroeconomic game between two groups of countries exercising fiscal policy and one joint central bank. The results shed light on a more realistic analysis of strategic scenarios for policy insights and finding optimal strategies, contrasting with traditional methods. In further developments on the Differential Evolution method application, Savin and Blueschke (2016) introduced a model to solve optimal control problems, addressing the limitations of classical methods in specific situations. The model's performance demonstrated a superior optimization of expected outcomes, strengthening the claim that heuristic methods are well-suited to navigate complex search spaces and find good approximations to global optima. Blueschke et al. (2020) later extended on this subject by introducing a novel Differential Evolution-based method for solving optimal control problems with passive learning. This learning method models the observation of the world's current state and employs new information to improve the system's general knowledge. The proposed approach does not imply the modification of the original problems and provides more robust results regarding the learning process.

Bullard and Duffy (1998) documented a macroeconomics experiment using a Genetic Algorithms-based learning model to simulate the behavior, outlining that a population of artificial agents can coordinate depending on the information structures they are inserted into, as chaotic and complex structures tend to hinder coordination. Another related 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. Where experimental data verify changes in actions, surroundings, and decision-making, such social trends remain resilient and difficult to alter, according to the study, offering insights into how to impact social development. Macedo et al. (2020) applied a Genetic Algorithm to optimize trading strategies, which outperformed the analyzed market indicators by employing a more comprehensive search space than traditional methods.

In a 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 reported how the devaluation of rhino horns would likely lead to higher poaching activity and that such an approach was only practical when combined with disincentives, intending to contribute to informing debates on the issue with scientific facts.

Arifovic and Ledyard (2011) introduced an evolutionary learning model with relatively good performance at matching the behavior of agents engaged in repeated strategic interactions when the behavior converges to a Nash equilibrium state. The authors state that most games do not require sophisticated strategies, except for the case of coordination games, which reduces the model's performance. Arifovic and Ledyard (2012) later reinforced the predictive power of evolutionary learning methods by introducing a comprehensive model able to generate data quantitatively similar to the empirical values, focused on the contribution mechanism of a public goods game. Price (1997)

also reported a good performance from Genetic Algorithms in searching for equilibria in standard games from industrial organization theory, such as Bertrand and Cournot competition scenarios.

Koza et al. (1992); Koza (1994) provided an early framework for Genetic Programming that introduced the notions of learning by modeling agents and their learning behavior over time. The author defines adaptive learning as the process of changing the structure of a potential solution. Hence, it performs better in its environment, where positive changes are rewarded and negative changes are discouraged by the underlying fitness function. Similarly, Chen et al. (2005) introduced a comprehensive game theory and Genetic Algorithms framework that approach several topics present in this research, such as coordination, adaptive learning, and equilibrium selection. The authors compared the behavior of computational agents to human subjects. They concluded that the behavior was remarkably similar in the applied experiment, supporting the idea of Genetic Algorithms as a credible tool to model human behavior.

Lensberg and Schenk-Hoppé (2021) studied the process of learning in one-shot multiple 2×2 games, where the agents never only see each game once and should learn to find optimal strategies based on information acquired across games. The author proposes a solution concept based on multiple artificial agents that learned how to play the games through Genetic Algorithms. The proposed theoretical model is reported to perform well, in line with intuition and 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 & Dorao, 2012, 2013; Mohamed & Koivo, 2011), communications (Kusyk et al., 2011), land usage (Liu et al., 2015), biology (Hamblin & Hurd, 2007) and ecology (Hamblin, 2013).

In summary, the literature suggests that Genetic Algorithms are an appropriate model for adaptive learning and optimizing strategic decisions. It performs well in problems of strategic interaction models (such as ours) while incorporating behavioral traits that are close to empirical findings in experiments with human behavior. Genetic algorithms suit our objective since they allow us to manipulate binary sequences under imposed constraints. In this case, one can transform the numerical structure of game elements so that essential characteristics are taken into account, as well as the outcomes of the decisions performed by the agents, expressed by the fitness function generated by different strategy selection rules. In this way, repeated decisions of the agents can influence the transformation of the games to directions that are consistent with the agents' behavior, providing us the necessary building blocks to simulate a situation where games can transform and analyze how the behavior of the agents influences these transformations.

3 Game-Theoretical Features

This article adopted the standard representation of strategic-form games as a model of simultaneous interaction between two agents, denoted by a 2×2 matrix. This form encompasses 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 (Von Neumann & Morgenstern, 1953; Robinson & Goforth, 2005). The strategic representation of games focuses on static analyses while overlooking dynamic aspects such as the order of the players' moves, changes in the moves, and the informational structure. This approach suggests the strategies that are more likely to be used by each player or alternatively recommend to 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 in the rows of the matrix and player *COL*'s strategies in the columns, respectively.

The adopted classes of 2×2 strategic-form games were based on the "periodic table" categorization provided in Robinson and Goforth (2005), which formally connects all ordinal rank games with distinct player preferences since swaps topologically linked the games in adjoining payoffs. The space of 2×2 is infinite, though as we were 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. 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 a broader representation of strategic scenarios, all 144 unique classes of games - including a wide range of well-known applied game theory situations such as the Prisoner's Dilemma, Chicken game, Stag Hunt, Battle of Sexes, and several others - are included in the simulation model.

In complement to the periodic table approach, Bruns (2010) further categorized the games by similarity in Layers, which outlines topological proximity, and Families, which groups similar games (based on equilibria and payoff structures, see Table 1). The games were split into populations following both grouping rules, as the scheme described in Fig. 1 and Table 1.

Fig. 1 Representation of the game families and layers in the periodic table of 2×2 games (Bruns, 2010, 2011, 2015a, b)

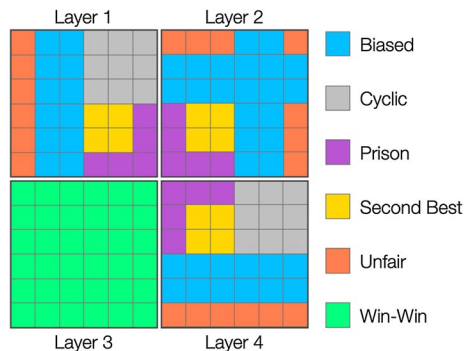


Table 1 Characteristics of games categorized in families (Bruns, 2015a, b)

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 get 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

4 Simulation Model Design

To have dynamic and transforming games, this simulation model processes the game-theoretical elements described earlier by allowing the games to change as a function of the players' strategic behavior. The Genetic Algorithms method becomes a fundamental building block, an enabler of these dynamic transformations. Genetic Algorithms are a class of search heuristic programs inspired by the process of natural evolution (Holland & Miller, 1991; Holland, 1992), being part of a broader set of comparable methods named Optimization Heuristics. The latter term, usually linked to algorithms inspired by nature, is defined by Gilli and Winker (2009) as methods that provide high-quality approximations to the global optimum, robust to changes, not too sensitive to parameters, easily deployable to many types of problems and might be stochastic, but without subjective elements. In Genetic Algorithms, the search space for potential solutions imitates the process of biological evolution. There are many variants of methods that are considered Genetic Algorithms. However, usually, this class of algorithms shares the following characteristics: having a population of individuals as potential solutions represented as binary strings, an objective function (fitness or cost), and the three types of genetic operators: selection, crossover, and mutation (Holland, 1992; Mitchell, 1998; Leszek, 2008; Slowik & Kwasnicka, 2020). The following sections will explain these individual elements in detail as we describe our implementation method.

Our approach allows the agents to modify their environment, especially regarding how they decide, to measure the progress of the genetic learning process in terms of strategy performance. The simulation process basically consists of playing the games in the designated populations and adopting one of three different strategy selection rules: (1) Nash equilibrium, (2) Hurwicz rule, and (3) random selection. After the strategies' selection, the games were assigned fitness scores based on the aggregated payoffs from both agents' choices; therefore, the games can be selected for replication in a way that favors higher-performing combinations of strategies. These preferences were implicitly expressed by the fitness function, which defined the quality of the selected strategies in terms of gained utility. In the next step, the games were processed by the Genetic Algorithm, which selects two games (parents) from the pool and generates a new game via the crossover and mutation operators. The new games were inserted back into the population, and the process was iterated a fixed number of times. See the high-level model overview in Fig. 2.

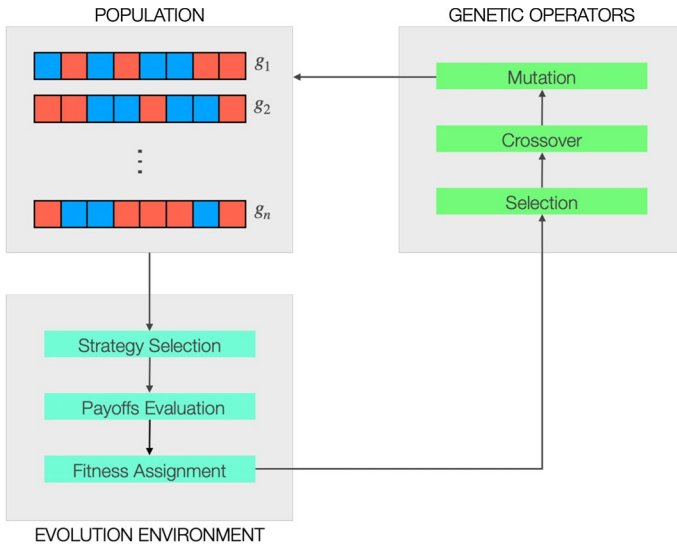


Fig. 2 Simulation model overview

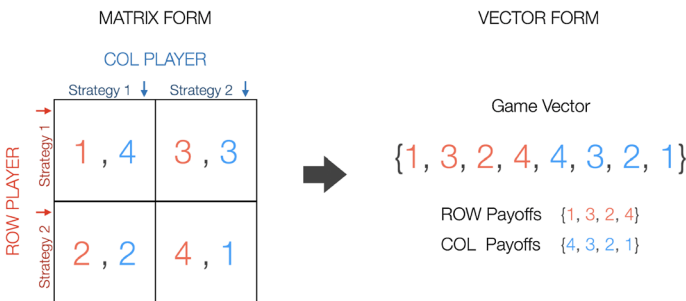


Fig. 3 Game vector encoding scheme

To manage the complexity and the processing requirements, the population numbers were kept constant throughout the iterations of the evolutionary model, which means when a new game is added, another game is excluded.

4.1 Populations and Data Structures

The encoding of games is based on a vector representation (payoff structure reproduced by integer vectors, as in Fig. 3). The analysis was rooted in two primary divisions of game pools based on families and layers. On the one hand, the first restriction derived from Robinson and Goforth (2005) ’s division of layers in their definition of the periodic table of 2×2 games, which considers all the 36 neighboring games according to the number of payoff swaps (see Fig. 1). The entire space (all 144 games, not allowing ties) was also handled as one distinct population, being

processed apart from the others, allowing us to analyze the population's results using the complete set of available characteristics in the pool.

On the other hand, the second restriction was based on Bruns (2015a)'s game families' categorization, split into six groups (as in Table 1). These two restrictions separated our groups of games into eleven distinct populations, processed and analyzed individually.

4.1.1 Binary Encoding

We adopted a binary representation of the vector form for the genetic operations. As our payoff matrices were represented by integers between 1 and 4, each possible payoff value has been encoded with one pair of bits. Consequently, the binary encoding was based on four possible sequences using a double binary representation:

$$\begin{aligned} 0, 0 &\rightarrow 1 \\ 0, 1 &\rightarrow 2 \\ 1, 0 &\rightarrow 3 \\ 1, 1 &\rightarrow 4 \end{aligned} \tag{1}$$

As an example of a vector transformed to binary, the representation of the Prisoners' Dilemma game (game in Fig. 3) is defined as:

$$[1, 3, 2, 4, 4, 3, 2, 1] \rightarrow [0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0] \tag{2}$$

This encoding scheme was applied in all 144 games. All game binary strings contained 16 bits, with each pair of bits representing one of the payoff values in the matrix. The game vectors could only be modified in accordance with this framework; thus, the structural scheme remained unchanged.

4.2 Evolution Environment: Strategy Selection and Fitness

This section explains the processes that govern the agents' strategy selection rules and how the players should act in the scenarios presented. The three approaches for selecting strategies were used to measure the impact of distinct types of strategic behavior.

4.2.1 Nash Equilibrium Strategy Selector

The Nash equilibrium is a strategy profile in which each strategy is an optimum response to other players' strategies. This logic holds to pure and mixed-strategy profiles (probability distribution over the available choices). As expected, utilities are linear in terms of probabilities. If a player in Nash equilibrium utilizes a non-degenerate mixed strategy, it must be indifferent from 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 (Von Neumann & Morgenstern, 1953; Fudenberg & Tirole, 1991; Maschler et al., 2013). In our games, we will

encounter all Nash Equilibrium situations, with strict and weak equilibria and pure and mixed strategies.

Because there can be multiple equilibria in bi-matrix games, the approach selected is the support enumeration method. (von Stengel, 2007; Widger & Grosu, 2008; Knight & Campbell, 2018). The support enumeration computes 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 $|J| = k_2$. In other words, the goal was to find support for strategies played with a non-zero probability. At this point, the algorithm evaluated the best response condition, ensuring no better utility residing outside of the supports.

The steps outlined in equations (3) and (4) iterate through all potential support pairs.

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

Sequentially, for considering mixed strategies:

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

There were two equilibrium-selection rules for games with more than one Nash equilibrium. The first rule is based on the payoff dominance concept for simulating self-maximizing behavior (Harsanyi et al., 1988). The equilibrium points might be finite or infinite, with at least one equilibrium (pure or mixed) for each game. Our structure identified three main configurations: pure-strict, pure-weak (when a stronger pure-strategy equilibrium was available), or mixed. The Nash equilibrium is payoff-dominant and henceforth selected if it is Pareto superior to all other possible equilibrium situations in a game. The second rule applied a simple random selection from the sample of computed Nash equilibria for each game. The objective was to enable the comparison of the maximization versus randomization approaches, and measure the effects in the evolutionary process generated by more diverse procedures for Equilibrium selection.

4.2.2 Hurwicz Rule Strategy Selector

This study's second strategy selection rule employed the Hurwicz criterion (Hurwicz, 1951). This rule introduces a coefficient of realism, α , which serves as a tool for balancing pessimism and optimism in decision-making under uncertain scenarios, allowing decision-makers to account for different possible outcomes. The pessimistic option employs the *maximin* criterion, while the optimistic option employs

the *maximax* criterion. The α parameter introduces a weighting factor between both extremes, simulating different degrees in behavior profiles.

Following Gaspars-Wieloch (2014)'s implementation, we applied the following formula.

$$h_j = \alpha \cdot w_j + (1 - \alpha) \cdot m_j, \quad (5)$$

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

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

This strategy selector introduced another model of behavior profiles and enriched the dataset to analyze simulation results. 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). For other applications of the Hurwicz criterion in decision-making under uncertain scenarios, 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 random choice of strategies for both players, simulating the total absence of strategic behavior. 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. The random selection was essential to outline the effects of strategic behavior introduced by the other two rules.

4.2.4 Payoffs and Fitness

Our fitness function reflected the players' preferences defined by the strategy selection rules applied at the game-playing 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 various available strategies. For systematic computations of payoffs, this algorithm employs a matrix multiplication method, using the dot product algebraic operation (Tanimoto, 2015). In this case, the strategies adopted by an agent during the execution of a game yielded probabilities distributions over the two possible strategies, expressed as the state vectors $(p_1^1(t), p_1^2(t))$, that is, the probability of player i selecting the strategy j (1 or 2) at period t .

We may depict the game (choice scenario) between two players by representing the reward matrix of the game structure as the matrix $ABCD$, the computation of the payoffs (π_1^j, π_2^j) at period t for a game as:

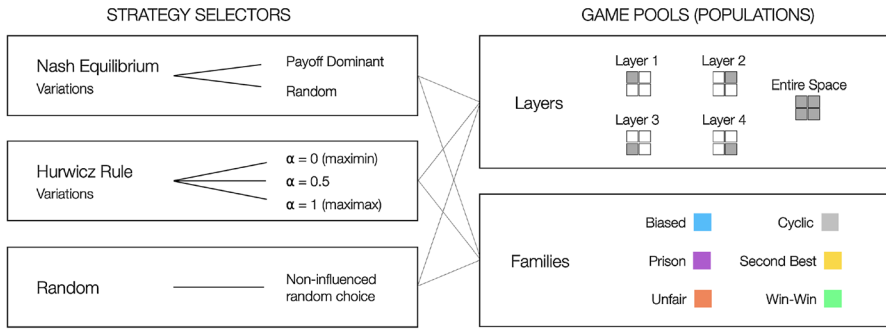


Fig. 4 Overview—Strategy Selectors versus Populations

$$\pi_1^j(t), \pi_2^j(t) = \begin{pmatrix} p_2^1(t) \\ p_2^2(t) \end{pmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot (p_1^1(t), p_1^2(t)) \tag{7}$$

This approach returns the matrix cell containing the expected payoffs (π_1^j, π_2^j) for players *ROW* and *COL*, respectively, at time t , 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:

$$f_g(t) = (\pi_1^j(t) + \pi_2^j(t))^2. \tag{8}$$

The acquired utilities were aggregated 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.

All variations of strategy selectors were applied to all game populations, as illustrated in Fig. 4. Once the strategies were selected, and the utility scores were assigned to the entire set of games, the game population was ready to be processed by the Genetic Algorithm environment.

4.3 Genetic Algorithms Implementation: Genetic Operators

The application of the Genetic Algorithms method in this study was designed to model the collective learning process within a population of games. Each individual (game) represented a search point in the space of all potential solutions for the introduced problems and a potential temporal container of current knowledge regarding the laws of the environment. The process starts by initializing a population, then it evolves towards enhanced performance regions of the search space utilizing randomized crossover, mutation, and selection processes, or genetic operators (Back, 1996). The optimization process in this context aimed to adjust the payoff structures in the games, to disseminate the best outcomes for the defined strategy selection rules, as described by Haupt et al. (1998).

4.3.1 Selection and Replacement

Selection is a crucial factor for directing the search process toward better individuals. According to Rowe (2015); Reeves and Rowe (2002), the fitness function assigns positive scores to well-performing games in the search space. To avoid early convergence, we opted for a non-deterministic approach, meaning that all individuals are considered with a certain probability during the selection process. Based on the experiments with distinct methods, we adopted the fitness proportionate selection, where the probability of a game g being selected is based on its performance against the rest of the population: $\frac{f(g)}{F}$. Where f is the game's fitness scores and F is the total fitness of the current population (aggregated).

The applied replacement method was based on the Inverse Selection criterion (Rowe, 2015), which links directly to the fitness function and the previously described selection method. The standard replacement rate chosen for the algorithm was 1. Since the population size remained constant, another game required removal whenever a new game was added to the current population. This rule made the poorer performing solutions more likely to be replaced than the better ones since the fitness determined the probability of being replaced.

4.3.2 Reproduction: Crossover and Mutation

The chosen reproduction method was a single-point crossover (Holland, 1992; Luca-sius & Kateman, 1993) that chooses a random index position 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 Fig. 5. The idea was to recombine building blocks (schemes) on different strings. The crossover operator ensured that new individuals inherited the parents' characteristics, likely to be high performers among the population.

As the next step, the mutation operator introduced a probability of changing one bit within the binary structure of an offspring game at a given generation. As illustrated in Fig. 5 (Right-hand side), if mutation takes place, one random bit will be

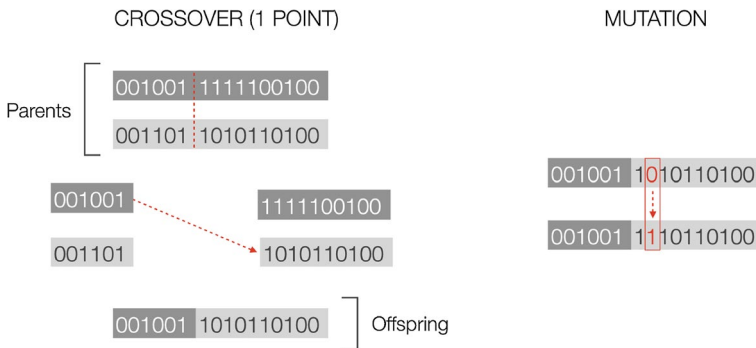


Fig. 5 Crossover and Mutation Operators Example

selected within the offspring game's (generated via crossover) binary string, with a uniform probability distribution over the possible 16 bits. The selected bit element is then flipped, from 1 to 0, or 0 to 1, generating a new game with new attributes, potentially not present in the parents' original gene pools. This step ensures the diversity of characteristics in the population and the possibility of generating more diverse games while avoiding fixation in determined sets of characteristics.

4.3.3 Evolutionary Loop

The outlined steps are iterated through multiple generations. The populations of games and their fitness scores were the main object of analysis for accessing the algorithm's properties and effects. The final data output consisted of a population of games with the same length as the initial set, outlining 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 expected to become constant when first reached, meaning that the algorithm found an optimal solution for the problem. As a recap, the model applied the following steps:

1. Initialization of 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. Execute the genetic operators: selection, crossover, and probably mutation within selected individuals in order to create a new game from the selected games.
5. Insert the offspring back into the population, replacing a game by inverse selection.
6. Return to step 2 and iterate this process until the termination criterion is satisfied

One important remark is that we employed a unified termination criterion for simplicity and experimental purposes at 10000 generations. Theoretically, the algorithm would have served its purpose when the entire game population reached the maximum fitness level (global optimum). We have recorded and documented the generation number in which all populations reach an optimal point as a measure of performance (speed of convergence) in Table 3. The shared termination threshold allowed straightforward comparisons of the datasets and the observation of the stability of equilibrium states over more extended periods. Similarly, for the mutation rate, we have applied a unified parameter value at 1%. The mutation rate affects the evolutionary process and speed of convergence. Since we analyzed several different combinations of populations and strategy selection rules, we kept a constant parameter for simplicity and comparability. However, we encourage practitioners to find optimal mutation rates for individual problems in optimization-based tasks through parameter fitting or experimentation. We have performed a few tests with different mutation rates in terms of convergence. An example of such tests is found in Appendix D.

Table 2 Summary: algorithmic parameters adopted in the model execution

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	<i>FPS</i>	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	Maximum number of generations (iterations) allowed

Apart from the termination criterion situation described above, the simulation model comes with a range of other distinct parameters to be defined by the practitioner. Table 2 summarizes the methods and parameters applied in this experiment.

5 Simulation Results

This study's concept of strategic behavior in decision-making was represented by variations of both the Nash equilibrium and the Hurwicz Rule strategy selectors. Alternatively, the absence of any strategic behavior was simulated by allowing the agents to select random strategies from the available options utilized as a control feature. The analysis of the experiment findings focused primarily on studying the effect of strategic behavior during the evolutionary learning process.

We quantified the influence of strategic behavior as the ability and speed of converging to optimal outcomes for an entire population, expressed by the individual utility levels. We applied the variations of the strategy selection methods into a range of different game pools, intending to assess this performance and also judging by the ability to transform the initial scenarios, which might be anything between pure conflict and harmony.

In addition to the utility convergence and strategies performance, the analysis of the simulation outcomes revealed two other interesting points, the coordination patterns in strategy selection by the agents and the transformation of the diverse game pools into sets of repeated games. Each of these points was explored more in-depth hereunder.

5.1 Utility Convergence

The simulation results indicated that the model mostly evolved in conformity with the reported literature and the expected development in accordance with the self-maximizing applied behavioral 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 overall tendency was that once an optimal strategy was found, there were no incentives

for the participants to deviate. Over time, successful tactics were generally replicated, resulting in 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 toward an optimal strategy. Only the rounds with the Nash and Hurwicz strategy selectors get the aforementioned results. The Random approach produced fuzzier and unordered payoffs, which were usually non-optimal.

The analysis of the utility development, that is, the payoffs gained as results of strategies selected, was plotted in Figs. 6, 7 and 8, for each of the populations and strategy selectors. Each plot's data divides into two combined plots displaying the utility development for each generic player type, *ROW*, and *COL*. The *Y* axes show the average utility levels for the whole population in each generation (*X* axes), using different colors for each population. Each figure shows the results for each of the strategy selector types, being Fig. 6 for the Nash equilibrium-based methods, Fig. 7

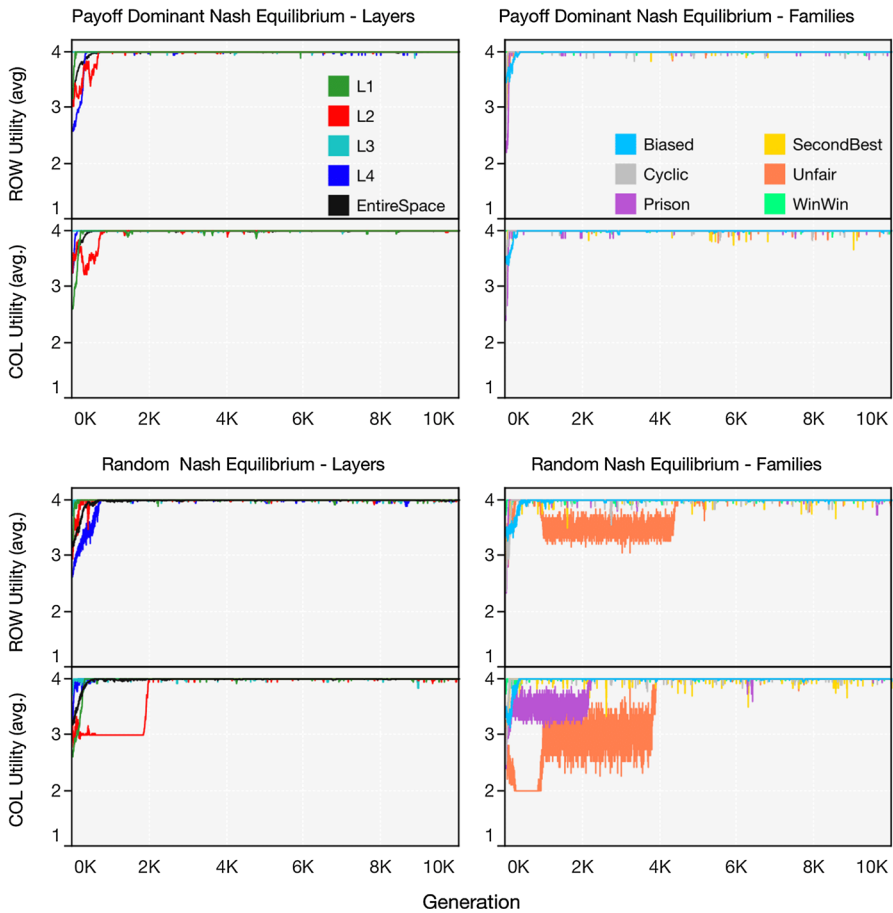


Fig. 6 Utility Development—Nash equilibrium Strategies

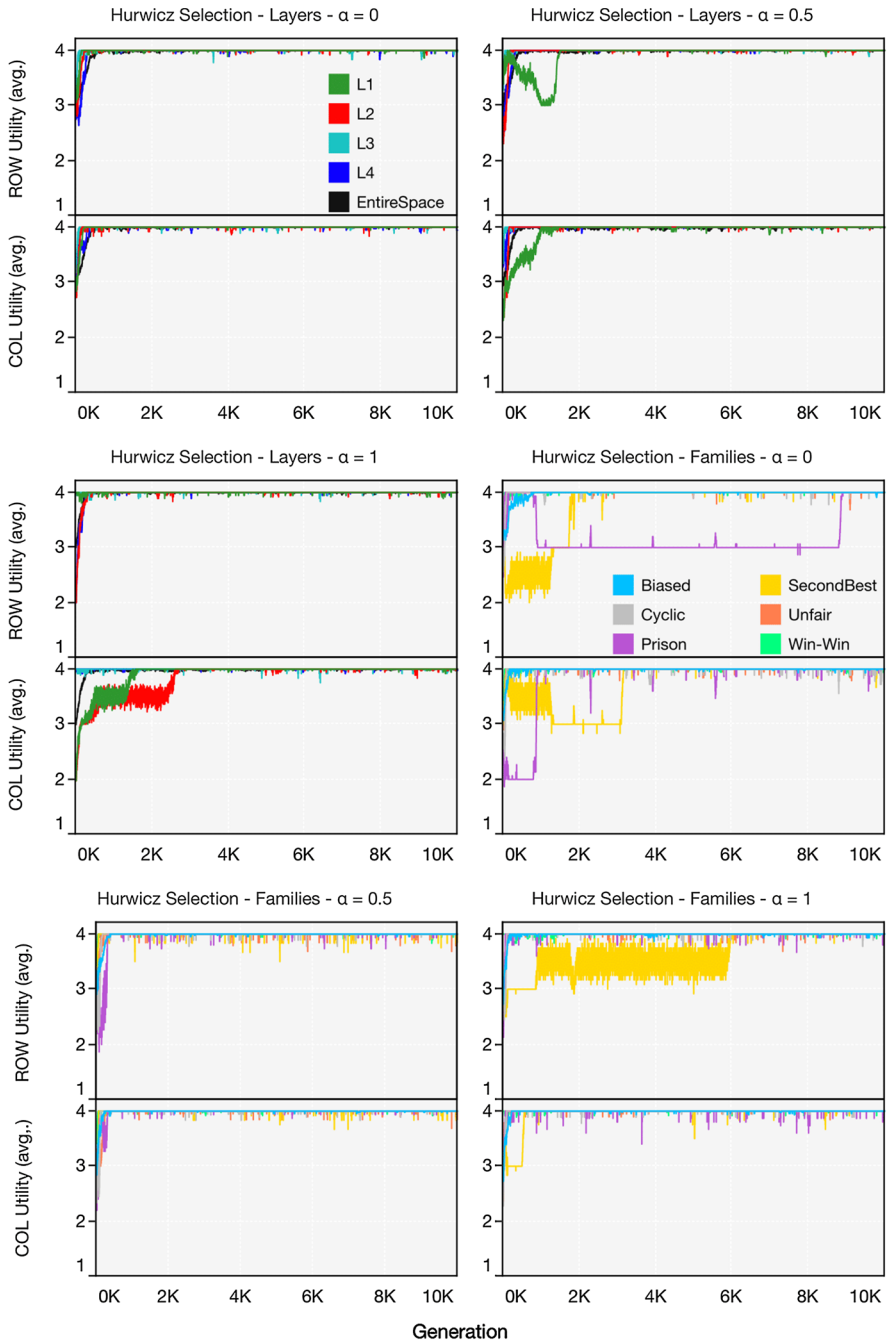


Fig. 7 Utility Development—Hurwicz rule Strategies

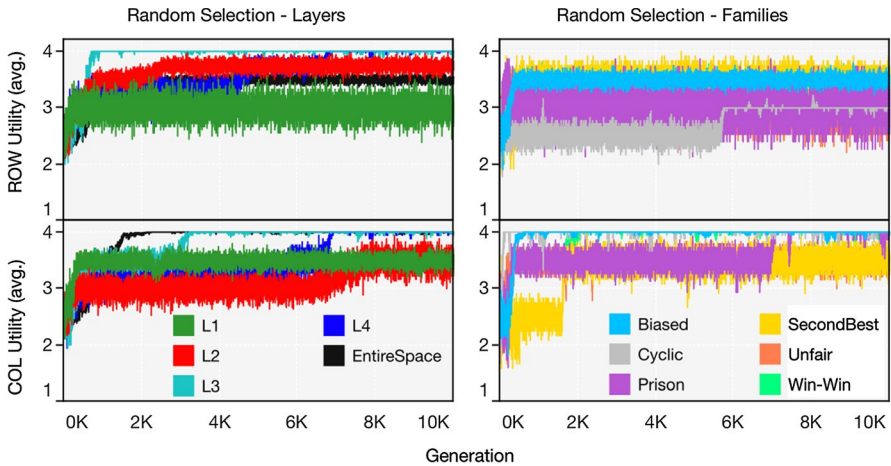


Fig. 8 Utility Development—Random Strategies

for the variations of the Hurwicz rule, and Fig. 8 for the completely random method. The populations are further divided into plots using the layers and family groups described earlier.

The evolutionary process gradually increased the individual utility to optimal values. The agents proceeded to select strategies that maximized the individual payoffs to maximize the overall outcomes of the games themselves, as defined by the fitness function. In this manner, the equilibrium became stable when all games in the given population achieved optimal payoffs. The selected strategies tended to persist and resist other invading strategies, as also observed by Friedman (1991). The charts displayed similar convergence curves in most cases.

Due to the stochastic mechanisms inherent to Genetic Algorithms, some games in the population may have lost an optimal pair of strategies during the evolutionary process. Nevertheless, the subsequent generations quickly adapted to the optimal strategies again. Similar effects of genetic experiments are observed in Riechmann (1998, 2001, 2002), where the learning process was given in two states: (1) the movement of populations towards a stable state, denominated behavioral 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. We observed similar trends in the equilibrium states observed in our results.

When looking at the Nash equilibrium runs (Fig. 6), one can observe that in all game pools, the Nash strategies could drive the convergence to the optimal utility values. Interestingly, the population that took the longest 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 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 Fig. 7) compared to the Nash equilibrium runs. 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. There was a more distinct evolutionary pattern in the families-based populations until reaching stability. The *Second Best* and *Prison* families demonstrated a much noisier and longer process to reach equilibrium when considering the optimistic and, even more so, the pessimistic approaches.

The random pools displayed an interesting control feature, exhibiting a noisier process, with marked lower populations that could not converge to the evolutionary stability, even after ten thousand iterations, for both layers- and families-based populations (Fig. 8). This lack of convergence properties is an essential contrast to the other runs, implying that the presence of strategic behavior highly influenced the evolutionary process itself. Even with a model tailored to favor individual payoffs, it was insufficient to drive the equilibrium state.

5.2 Strategies Performance

This section analyses the strategy selectors' performance in terms of convergence speed to optimal values. Table 3 summarizes the speed of convergence results, that is, the number of generations it took for the entire population to reach maximum fitness scores. The payoff dominant Nash equilibrium strategy displays the best overall performance, followed by the Hurwicz rule with $\alpha = 0.5$, which alternates between the maximin and maximax strategies according to the game structures. In the third place, we have the Hurwicz rule with $\alpha = 1$, or maximax (optimistic), followed by the Nash equilibrium with random equilibrium selection, and the Hurwicz rule with

Table 3 Rank of strategies performance in each of the distinct populations

Population	NE: pdom eq.	NE: rand eq.	HR $\alpha = 0$	HR $\alpha = 0.5$	HR $\alpha = 1$	Random
Biased	363	401	737	392	217	failed
Cyclic	70	149	90	161	89	failed
Prison	95	2160	8881	298	54	failed
SecondBest	53	101	3148	55	4549	failed
Unfair	116	4382	73	235	58	failed
Win-Win	1	27	33	40	1	failed
L1	208	316	226	1494	1491	failed
L2	828	1972	209	178	2600	failed
L3	1	23	81	39	1	3201
L4	355	681	432	319	271	6842
Entire Space	701	653	742	709	612	failed
Average	254	988	1332	356	904	n.a.
Min	1	23	33	39	1	n.a.
Max	828	4382	8881	1494	4549	n.a.
Std. Deviation	283	1349	2658	424	1456	n.a.
Rank (avg.)	1	4	5	2	3	6

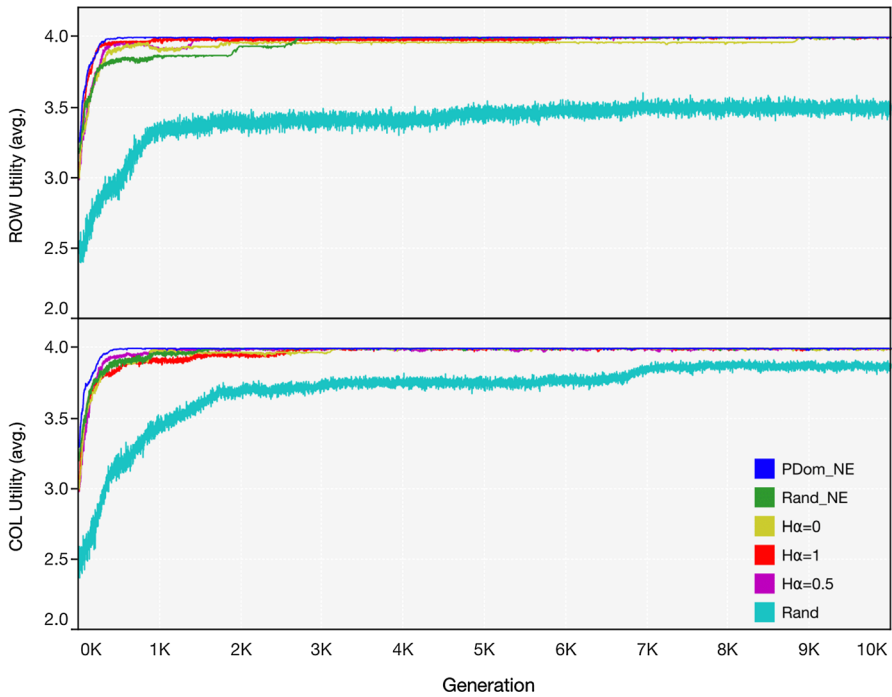


Fig. 9 Strategies Performance—Consolidated

$\alpha = 0$, or maximin (pessimistic). As denoted in the previous chapter, one can notice the lack of convergence for the random strategy selection method. In most cases, the random selection failed to converge within the allowed time, reinforcing the influence of strategic behavior simulation in the process.

Upon analyzing the game pools separately, we observed that pools with a higher number of conflict and non-optimal types of games, such as *Biased*, *Unfair*, *Second-Best* and *Prison*, affect the speed of convergence as well, requiring more iterations in order to reach optimized fitness values. Although the games will be transformed in every case and eventually converge to optimal scenarios when in the presence of strategic behavior, the available population characteristics directly influence how the games will transform in the next generations.

The charts in Fig. 9 demonstrate the utility development for both players by adopting each strategy selector and variations used in the simulation model, aggregated by average across all populations. Here it is easy to understand this rank and compare the random method against the others. The performance of the strategies varied on a game basis; a complete overview of each combination of game pool and strategy selector is presented in Appendix A.

5.3 Evolutionary Stability: Coordination

Following our game structure, each agent could select between two strategies, generically denominated as *Strategy1* and *Strategy2*. Additionally, for the Nash equilibrium-based pools, mixed strategies were allowed. The frequency of selected strategies based on a random sample of populations is presented in Fig. 10. One can notice that the evolutionary stability, in terms of utility, is reflected directly in the stability of the strategies adopted by each player. This observed trend was in line with the examples detailed in Weibull (1997); Hines (1987). Furthermore, most displayed a symmetric plot, especially after reaching equilibrium.

The literature suggests that in equilibrium situations, the players coordinate their strategies. 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 allows the equilibrium state to persist throughout the generations (see Fig. 10).

In addition, due to their inferior outcomes, mixed strategies have been eliminated early in the evolutionary process. Figure 10 displays a random sample of six strategy selectors and game pool combinations. The complete overview of the frequency of the selected strategies in all pools is presented in Appendix B.

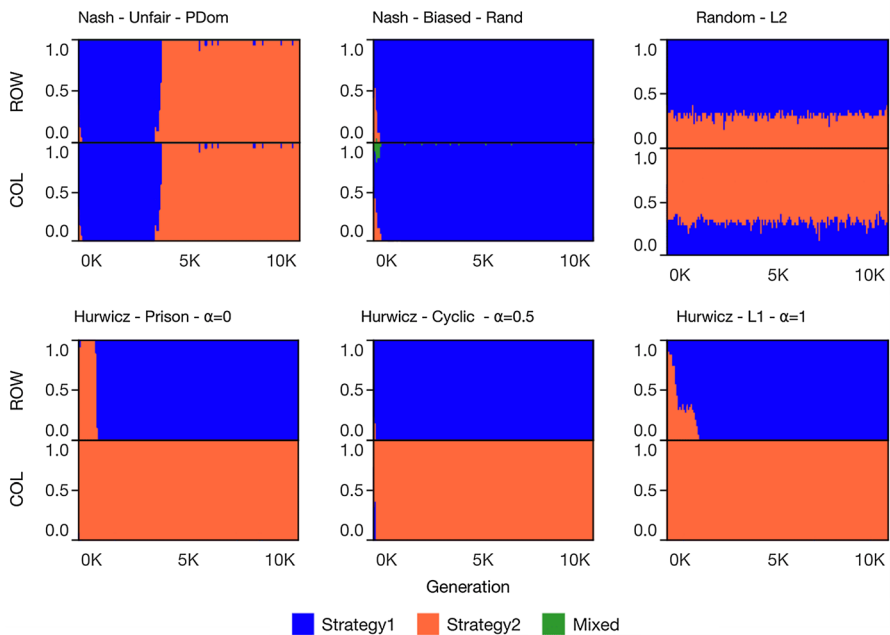


Fig. 10 Evolutionary Stability—Sample Pools (percentage of strategies used in each round)

5.4 Transformation of Games

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 infinitely many) Nash equilibrium (12.5% of the games). Table 4, depicts the resulting equilibria 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.

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 was only sometimes 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 (harmonious) game family.

The analysis of the results uncovered another interesting fact. Most games transformed and replicated one-selves, reducing the number of unique games in the final populations, where successful games appeared repeatedly. There was no restriction to how the game may change, except for the rules defining the periodic table of 2×2 games (Robinson & Goforth, 2005). Even when the payoff symmetry was broken, both players had an optimal equilibrium state, which tended to survive across generations. The games were changed in such a way that they retained (in the majority of cases) the strategy choices, and payoff yield stabilized according to the characteristics of the initial populations and the decision rules applied. In this case, the games were specifically optimized to create favorable decision scenarios for both players.

This point is visualized in Fig. 11, 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 few different games, repeatedly occurring within the same pool. The complete overview of the count of games in all pools is found in appendix C.

5.5 Comparison with Reinforcement Learning

Reinforcement Learning is a broadly applied concept in the economics literature, especially when it comes to game theory and situations of strategic interaction, as a model of the human learning process, as it is regarded as an alternative method to this paper for modeling learning behavior. A popular version of the

Table 4 Frequencies of the equilibrium structure for games in the final populations, compared to the initial pools

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%

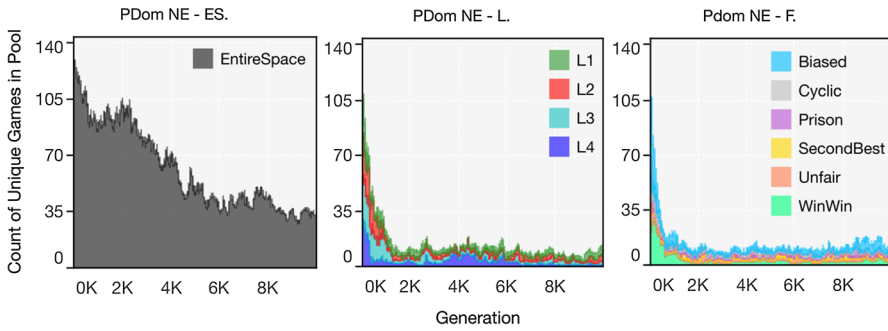


Fig. 11 Count of unique games in pool across generations—Sample pools

model was introduced by Erev and Roth (1998), in which payers change their choice probabilities in reaction to payoffs from previous rounds. Duffy (2006) documented comparisons between Genetic Algorithms and Reinforcement Learning in the context of economics by mentioning two examples. In the first comparison, Haruvy and Ünver (2002) analyzes a scenario based on a procurement-type market experiment, where buyers and sellers are supposed to reach a stable outcome. Applying Reinforcement Learning and Genetic Algorithms yielded similar predictions consistent with empirical evidence. In the second example, Arifovic and Ledyard (2004) compare both methods' performances in the context of a public goods game, also employing a reinforcement-belief hybrid model known as Experience-Weighted Attraction (Camerer & Hua Ho, 1999). By measuring performance in terms of time taken to converge to an equilibrium state (same rule used in our paper), the authors conclude that Reinforcement Learning ranks significantly worse than the other two methods. In contrast, the Genetic Algorithm displays the best performance.

As the model introduced in this paper is highly customized, a comparison might be beneficial for the reader to understand the discussion about performance and speed of convergence. For this reason, we applied a version of Erev and Roth (1998)'s model to the same pools of games and identically documented the results in terms of utility optimization and strategy selection. However, it is of utmost importance to understand that in our original model, we allow the Genetic Algorithm to modify the games in a population. In contrast, the Reinforcement Learning model can only learn optimal strategies restricted by the static structure of the games, which will not change over time. In other words, as the programs start iterating through the data, they will eventually be processing different sets of games, even though the starting pools are the same.

Details on how we implemented the Reinforcement Learning algorithm can be found in Appendix E. The model implemented for the comparison takes two parameters for a better fit to the data: ϕ , denoting the “forgetting rate,” that is, how quickly the agents forget past payoffs, and λ , which defines the sensitivity to the weights assigned to strategies for the generation of choice probabilities. For simplicity, we applied static parameter values for all game pools, found through over

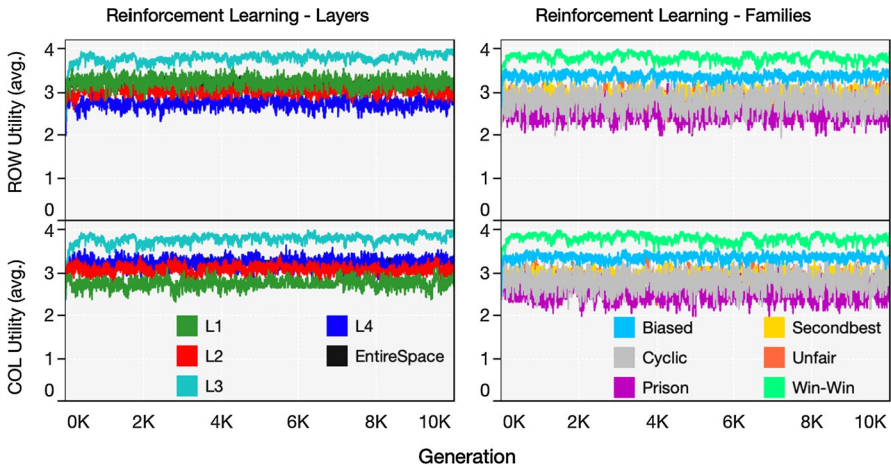


Fig. 12 Reinforcement Learning Algorithm Application

1000 simulations of 1000 rounds each. We calculated the average values of the pair of parameters that were most frequent in optimized outcomes in each population. Based on the results, the global parameter values were set at $\phi = 0.42$ and $\lambda = 0.9$, and initial attractions were set to 0; that is, no previous experience is assumed for the agents in any of the games.

Figure 12 displays the results for the Reinforcement Learning application. As suggested by earlier comparisons, Reinforcement Learning (and many other similar models), by nature, find suitable strategies that maximize the rewards in a given scenario. The fitness values converged to local optimum values. However, the main difference in our setting is that the Reinforcement learning model is constrained to the structure of the games, as observed in the summary of the results. The utility will never reach a global optimum if optimal-yielding strategies are not available for selection, as the games remain static over time.

When analyzing the strategy selection behavior of the agents, we noticed weaker coordination of strategies using reinforcement learning, as demonstrated in Fig. 13. Agents learned optimal strategies within each of the games. However, no consistent optimal-yielding strategies can be selected across an entire population, outlining the Genetic Algorithm's capacity to optimize multiple scenarios by transforming individual situations. Therefore this higher coordination effect cannot be achieved in optimizing multiple static games, and it shows similar figures to the random strategy selection method introduced before within the Genetic Algorithm implementation.

Fundamentally, both methods are based on similar premises but have different operating processes. Reinforcement Learning methods, such as the one used for this comparison, proved to be very efficient in finding optimal payoff-maximizing strategies, even using global parameters. However, the Genetic Algorithm performs better in our environment as strategic behavior rules allow the algorithm to change the

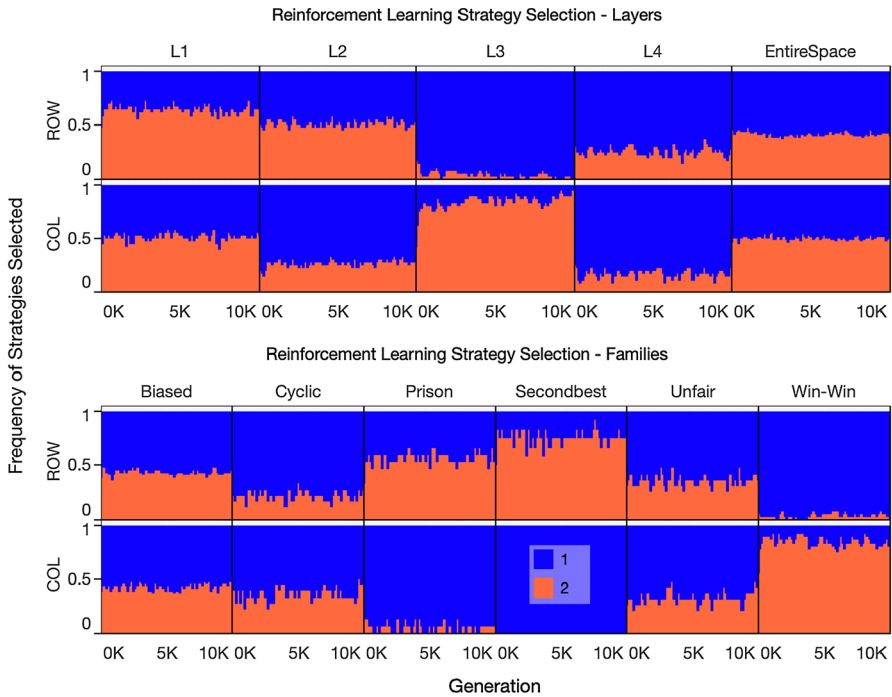


Fig. 13 Reinforcement Learning Strategies Selection Overview

games based on well-performing strategies. This ability enables the transformation of the games based on the agents’ behavior, which optimizes the decision environment as a whole, introducing the possibility of new, potentially better strategies.

6 Conclusion and Discussion

This paper introduced a novel game-theoretical approach for analyzing dynamic scenarios that uses a heuristic approach to transform games played sequentially based on the decisions performed by the agents, taking inspiration from observed scenario transformations over time. Based on how economic agents learn and transform their decision environments in the real world, our motivation was to document and describe the process of implementing a Genetic Algorithm as a mechanism for evolutionary economic learning. For this purpose, we compared the convergence rate to optimal scenarios of different strategy selection rules in different populations of diverse 2×2 game types. The presented analysis demonstrated how strategic

behavior can influence this dynamic learning process and how different strategic behavior mechanisms perform.

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

1. The simulated strategic behavior for decision-making directly impacted the evolutionary process, being responsible for transforming games into optimal scenarios (from conflict to harmony).
2. Similarly, the absence of strategic behavior negatively affects the Evolutionary process, preventing the evolution toward optimized scenarios.
3. The rank of the strategy selection methods from best to worse performance is given as follows: (1) payoff dominant Nash equilibrium, (2) Hurwicz rule with $\alpha = 0.5$, (3) Hurwicz rule with $\alpha = 1$, (4) Nash equilibrium with random equilibrium selection, (5) Hurwicz rule with $\alpha = 0$ and (6) random strategy selection.
4. When agents maximize their utility (i.e., behave strategically), the games evolve to have higher payoff structures, optimizing their decision environments and transforming conflict into harmonious games.
5. The evolution of dynamic strategic situations tends to create games where coordination is possible.
6. Decision rules and the environment characteristics influence the optimization process's agility in reaching an evolutionarily stable equilibrium.
7. Comparisons with reinforcement learning outline the power of transforming the games for the evolution of conflict to optimal scenarios. The reinforcement learning model can find optimal strategies within the game constraints but cannot optimize further without structural changes in the population of games.

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

The introduced decision rules presented a satisfactory performance in supporting the transformation of the games in 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 ranked similarly. However, in essence, the Nash equilibrium was still found to be, in this study, the most rapid and robust method for optimization modeling, in conformity with past findings by Sefrioui and Perlaux (2000).

The self-maximizing behavior and the transformation of the games allowed the individual strategies to evolve so that the agents tended to coordinate their strategies. Hence, for each player's strategies, an optimal best response is consistent over time. 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), the rational learning

in replicated games eventually leads to stationary points of the Nash equilibrium. Such a result is also in line with the concept of genetic stability documented in Riechmann (1998, 2001, 2002). In essence, strategic behavior allowed the agents to transform each game scenario into mutually optimal situations, eliminating conflicts and inequality.

Per Savin et al. (2018), incorporating the dynamics of strategic scenarios in equilibrium analysis allows for more realistic interpretations of possible environmental transformations and policy outcomes. Compared to this paper, our results relied on transforming strategic scenarios due to strategic behavior. We also conclude that heuristic techniques and their extensions provide flexible tools with broader search spaces, potentially resulting in superior and even innovative outcomes than traditional procedures. The comparison with Reinforcement Learning in our research also outlines the limitations of commonly employed methods, proposing new solutions to more flexible problems and providing insights into reality-inspired dynamics.

As discussed by Erev and Roth (1998); Brown and Rosenthal (1990); Chmura et al. (2012), the actual human behavior is at times not well predicted by the standard theory. However, in reality, the evolutionary social models are more complex and require significant efforts to create representative pictures that capture relevant characteristics of social systems (Reschke et al., 2001). The model created in this study aimed at an exploratory analysis that contributes to the discussion on how to envision and build representative models of strategic behavior and economic learning in different situations, exploring the effects of decisions in repeated strategic interaction.

We developed our simulation model to be flexible and integrate other ideas that enable the examination of dynamic games, strategic behavior, and economic learning in the future. Further research shall be performed by enriching the current model with different types of games (also diverse populations) and encoding different strategy selection models, which should capture more diverse behavioral profiles of decision-makers in varying contexts. In another exploration direction, 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. Furthermore, different Evolutionary Programming methods can be compared with the results achieved with the Genetic Algorithms application.

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

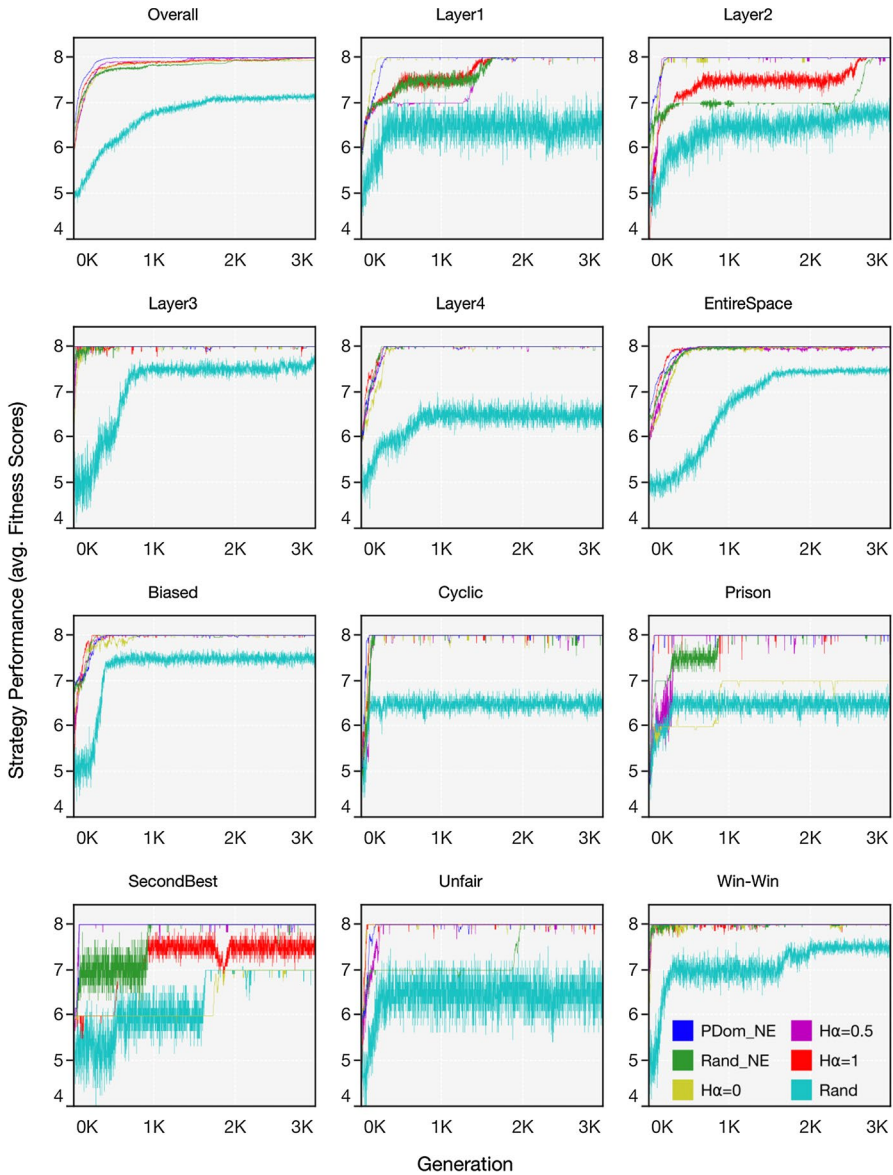


Fig. 14 Strategies performance—All pools

Appendix B: Complete Diagram of Evolutionarily Stable Strategies

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

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

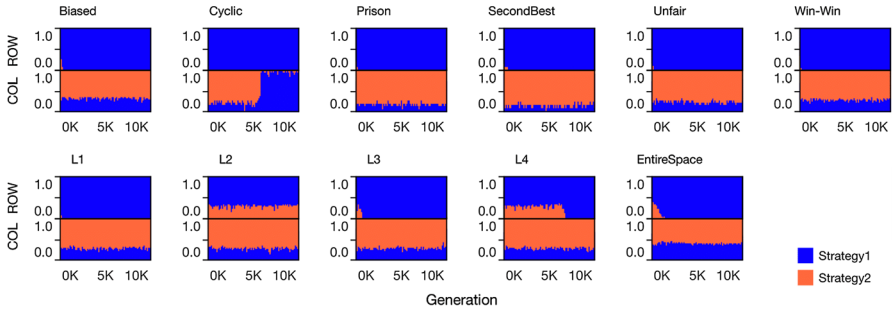


Fig. 15 Evolutionary stability—Nash pools

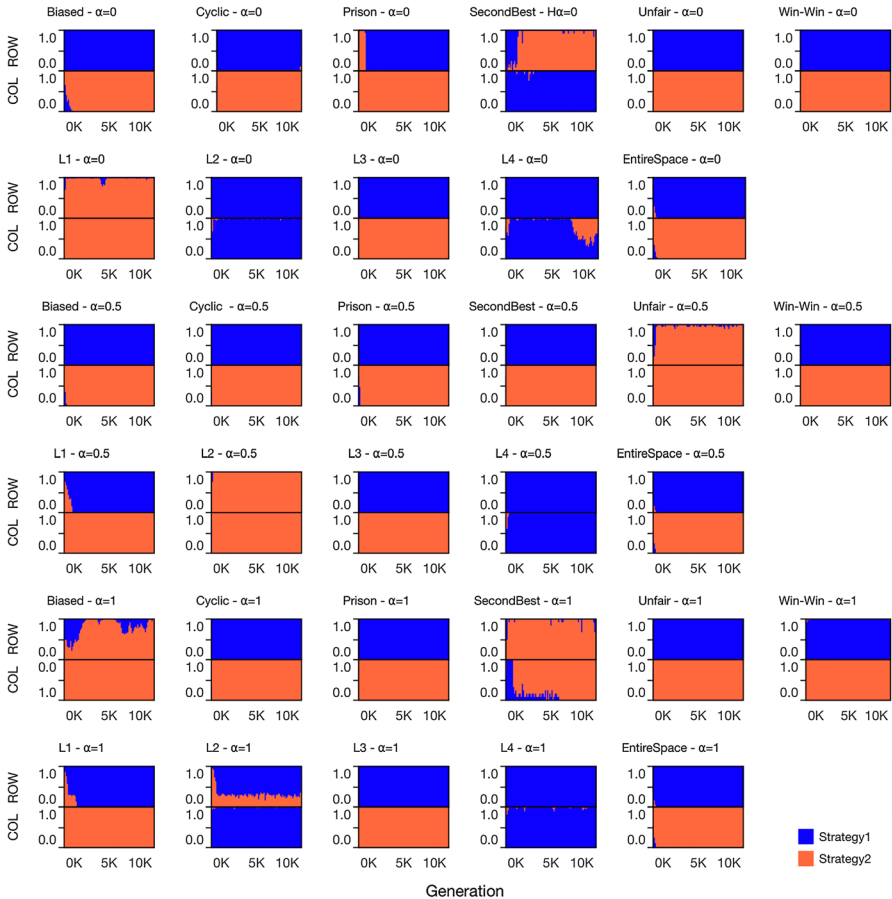


Fig. 16 Evolutionary stability—Hurwicz pools

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

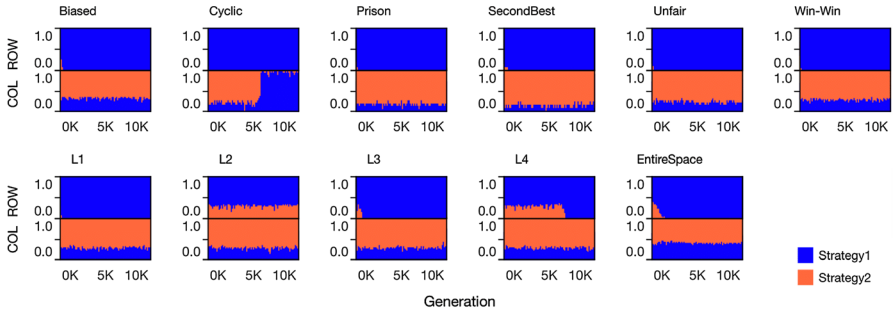


Fig. 17 Evolutionary stability—Random pools

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.

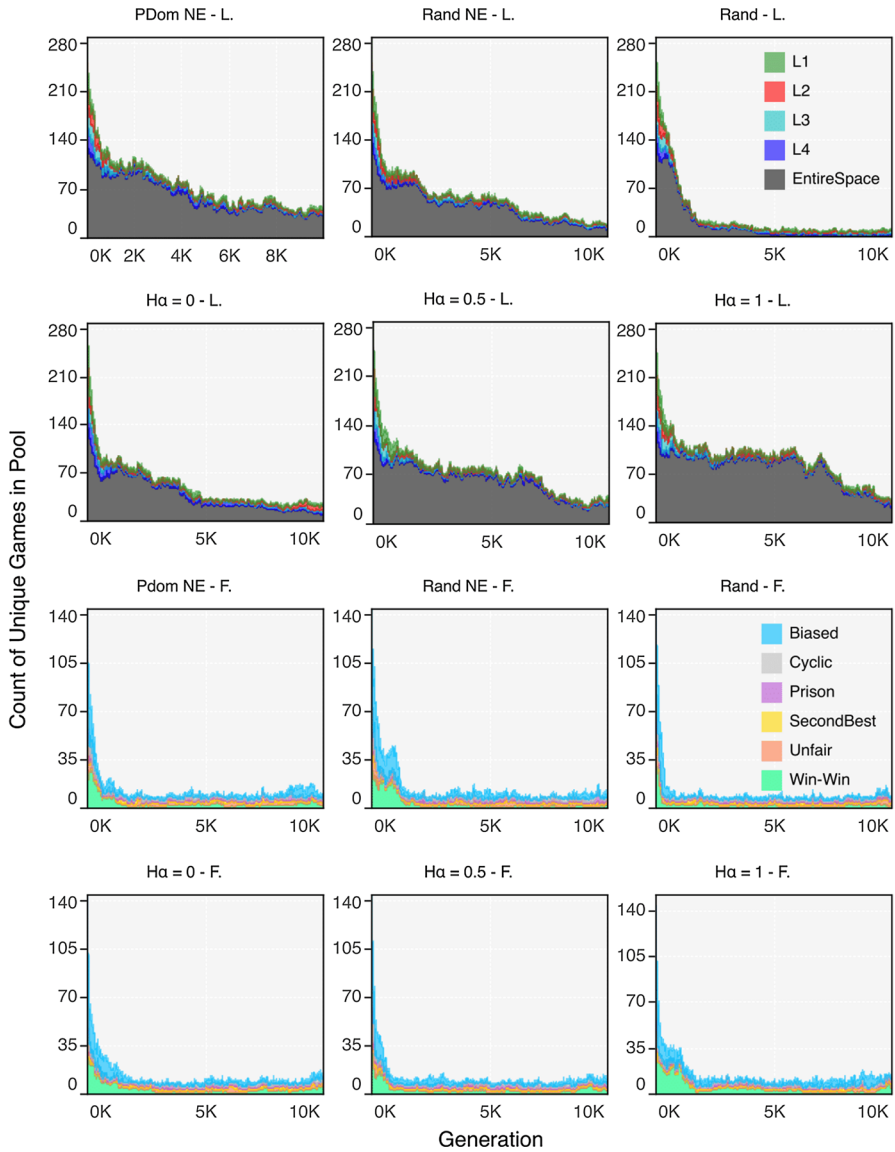


Fig. 18 Count of unique games in pool across generations—All pools

Appendix D: Mutation Rate Tests

Figure 19 summarizes the tests performed with mutation rates using the Payoff Dominant Nash Equilibrium strategy selector, which is the highest performing and most robust of our rules, applied in the whole pool of 144 games.

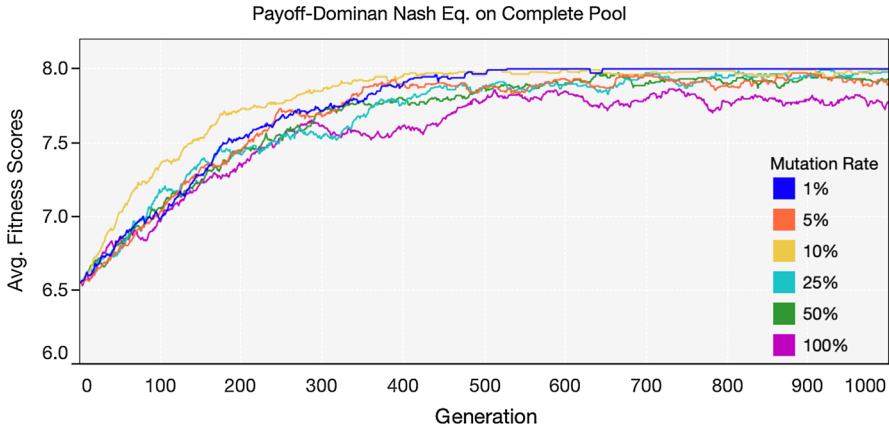


Fig. 19 Mutation rate tests—Payoff-Dominant Nash Eq. applied on the complete pool of games

Appendix E: Reinforcement Learning Algorithm

The reinforcement learning algorithm applied for a comparison with the Genetic Algorithms was based on Erev and Roth (1998) ’s model and in the formulation presented in Camerer and Hua Ho (1999); Moffatt (2020), which introduced the concept of attractions A_i^j , as weights attached to strategies. Initial attractions $A_i^j(0)$ can be given values to assume a degree of previous experience from the agents with the game (having to be estimated by the practitioner), or they can be neutral by setting default values to 0. The generalized model has two main components. First, the attractions’ update function at time t $A_i^j(t)$, given as:

$$A_i^j(t) = \phi A_i^j(t - 1) + I(s_i^j, s_i(t))\pi_i(s_i^j, s_{-i}(t)), \tag{E1}$$

where $\pi_i(s_i(t), s_{-i}(t))$ is the player i ’s payoff in round t (scalar-valued function) and I is the indication function, taking the value of 1 if the statement is true and 0 otherwise. This means that in reinforcement learning, a player’s attraction to a strategy can only increase if that strategy is chosen. ϕ indicates the speed at which past pay-offs are forgotten. $\phi = 0$ would indicate that only the most recent payoff is remembered. $\phi = 1$ would indicate that all past payoffs have equal weights in the current decision.

The second component is the transformation of attraction into choice probabilities via logistic transformation, given as:

$$P_i^j(t + 1) = \frac{e^{\lambda A_i^j(t)}}{\sum_{k=1}^{m_i} e^{\lambda A_i^k(t)}} \tag{E2}$$

Where $P_i^j(t + 1)$ is the probability of player i playing strategy j in round t and m_i is the number of strategies player i has. The parameter λ defines the sensitivity to attractions; attractions are irrelevant if $\lambda = 0$, and attractions are important if λ is

large. The choice probabilities are then used as means of selecting strategies in each round.

Funding Open Access funding enabled and organized by Projekt DEAL. This project received no external funding

Data Availability The source code and dataset examples generated and analyzed in this paper are available to download via the CoMSES platform at: <https://doi.org/10.25937/smg0-0t92>.

Declaration

Conflict of interest All authors declare that they have no conflicts of interest

Ethical Approval This paper does not contain any studies with human participants or animals performed by any of the authors

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Arifovic, J., & Ledyard, J. (2004). Scaling up learning models in public good games. *Journal of Public Economic Theory*, 6(2), 203–238.
- Arifovic, J., & Ledyard, J. (2011). A behavioral model for mechanism design: Individual evolutionary learning. *Journal of Economic Behavior & Organization*, 78(3), 374–395.
- Arifovic, J., & Ledyard, J. (2012). Individual evolutionary learning, other-regarding preferences, and the voluntary contributions mechanism. *Journal of Public Economics*, 96(9–10), 808–823.
- Axelrod, R. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration* (Vol. 3). Princeton University Press.
- Axelrod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. *The Dynamics of Norms*, 1, 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., & Scheffran, J. (2006). Agent-based computational modelling: An introduction. Agent-based computational modelling (pp. 1–16). Springer.
- Blueschke, D., Savin, I., & Blueschke-Nikolaeva, V. (2020). An evolutionary approach to passive learning in optimal control problems. *Computational Economics*, 56(3), 659–673.
- Brown, J.N., & Rosenthal, R.W. (1990). Testing the minimax hypothesis: A re-examination of o'neill's game experiment. *Econometrica: Journal of the Econometric Society* 1065–1081.
- Bruns, B. (2010). Navigating the topology of 2x2 games: An introductory note on payoff families, normalization, and natural order. arXiv preprint [arXiv:1010.4727](https://arxiv.org/abs/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. (2015). Atlas of 2x2 games: Transforming conflict and cooperation.

- Bruns, B. (2015). Names for games: Locating 2×2 games. *Games*, 6(4), 495–520.
- Bullard, J., & Duffy, J. (1998). Learning and the stability of cycles. *Macroeconomic Dynamics*, 2(1), 22–48.
- Camerer, C. (2003). Strategizing in the brain. *Science*, 300(5626), 1673–1675.
- Camerer, C. (2011). *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press.
- Camerer, C., & Hua Ho, T. (1999). Experience-weighted attraction learning in normal form games. *Econometrica*, 67(4), 827–874.
- Camerer, C., & Weigelt, K. (1988). Experimental tests of a sequential equilibrium reputation model. *Econometrica: Journal of the Econometric Society* 1–36.
- Castillo, L., & 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., & Dorao, C. A. (2012). Consensual decision-making model based on game theory for Ing processes. *Energy Conversion and Management*, 64, 387–396.
- Chen, S.-H., Duffy, J., Yeh, C.-H. (2005). Equilibrium selection via adaptation: Using genetic programming to model learning in a coordination game. *Advances in dynamic games* (pp. 571–598). Springer.
- Chmura, T., Goerg, S. J., & Selten, R. (2012). Learning in experimental 2×2 games. *Games and Economic Behavior*, 76(1), 44–73.
- Chmura, T., Kaiser, J., & 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.
- Dawid, H. (1999). Adaptive learning by genetic algorithms, analytical results and applications to economic models.
- Dawid, H., & Kopel, M. (1998). On economic applications of the genetic algorithm: A model of the cobweb type. *Journal of Evolutionary Economics*, 8(3), 297–315.
- Duffy, J. (2006). Agent-based models and human subject experiments. *Handbook of computational economics*, 2, 949–1011.
- Erev, I., & Roth, A.E. (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American economic review* 848–881.
- Fogel, D. B. (1998). *Artificial intelligence through simulated evolution*. Wiley-IEEE Press.
- Fogel, D. B. (2000). What is evolutionary computation? *IEEE Spectrum*, 37(2), 26–32.
- Freedman, L. (1998). *The revolution in strategic affairs*. USA: Oxford University Press.
- Freedman, L. (2017). The transformation of strategic affairs (No. 379). Routledge.
- Friedman, D. (1991). Evolutionary games in economics. *Econometrica: Journal of the Econometric Society* 637–666.
- Fudenberg, D., & 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.
- Gilli, M., & Winker, P. (2009). Heuristic optimization methods in econometrics. *Handbook of computational econometrics* 81–119.
- Gintis, H., et al. (2000). *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton university press.
- Glynatsi, N. E., Knight, V., & Lee, T. E. (2018). An evolutionary game theoretic model of rhino horn devaluation. *Ecological Modelling*, 389, 33–40.
- Goldberg, D. E., & 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., & Hurd, P. L. (2007). Genetic algorithms and non-ess solutions to game theory models. *Animal Behaviour*, 74(4), 1005–1018.
- Harsanyi, J.C., Selten, R. (1988). A general theory of equilibrium selection in games. *MIT Press Books* 1.
- Haruvy, E., & Ünver, M.U. (2002). Equilibrium selection in repeated b2b matching markets”.

- Haupt, R. L., Haupt, S. E., & Haupt, S. E. (1998). *Practical genetic algorithm* (Vol. 2). New York: Wiley.
- Hayashi, N., Ostrom, E., Walker, J., & 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* 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.
- Holland, J. H., & Miller, J. H. (1991). Artificial adaptive agents in economic theory. *The American Economic Review*, *81*(2), 365–370.
- 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., & Paris, C. (2007). Information processing under imprecise risk with the hurwicz criterion. 5th international symposium on imprecise probability: Theories and applications (pp. 233–242).
- Kalai, E., & Lehrer, E. (1993). Rational learning leads to nash equilibrium. *Econometrica: Journal of the Econometric Society* 1019–1045.
- Knight, V., & 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.
- Koza, J. R. (1994). *Genetic programming ii: Automatic discovery of reusable programs*. MIT press.
- Koza, J.R. (1992). Evolution of subsumption using genetic programming. *Proceedings of the first European conference on artificial life* (pp. 110–119).
- Kusyk, J., Sahin, C. S., Uyar, M. U., Urrea, E., & 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.
- Lensberg, T., & Schenk-Hoppé, K. R. (2021). Cold play: Learning across bimatrix games. *Journal of Economic Behavior & Organization*, *185*, 419–441.
- Leszek, R. (2008). *Computational intelligence: Methods and techniques*. Springer Press.
- Liu, Y., Tang, W., He, J., Liu, Y., Ai, T., & 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., & Kateman, G. (1993). Understanding and using genetic algorithms part 1. Concepts, properties and context. *Chemometrics and Intelligent Laboratory Systems*, *19*(1), 1–33.
- Macedo, L. L., Godinho, P., & Alves, M. J. (2020). A comparative study of technical trading strategies using a genetic algorithm. *Computational Economics*, *55*(1), 349–381.
- Manson, S. M. (2006). Bounded rationality in agent-based models: Experiments with evolutionary programs. *International Journal of Geographical Information Science*, *20*(9), 991–1012.
- Maschler, M., Solan, E., & Zamir, S. (2013). *Game theory*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511794216>
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Moffatt, P. (2020). *Experimetrics: Econometrics for experimental economics*. Bloomsbury Publishing.
- Mohamed, F. A., & Koivo, H. N. (2011). Multiobjective optimization using modified game theory for online management of microgrid. *European Transactions on Electrical Power*, *21*(1), 839–854.
- Pažek, K., & 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., & 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., & 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. (2005). Genetic action trees a new concept for social and economic simulation (Tech. Rep.). Germany: University Library of Munich.
- Plous, S. (1993). The nuclear arms race: Prisoner's dilemma or perceptual dilemma? *Journal of Peace Research*, 30(2), 163–179.
- Price, T. C. (1997). Using co-evolutionary programming to simulate strategic behaviour in markets. *Journal of Evolutionary Economics*, 7(3), 219–254.
- Puerto, J., Mármol, A., Monroy, L., & Fernández, F. (2000). Decision criteria with partial information. *International Transactions in Operational Research*, 7(1), 51–65.
- Rechenberg, I. (1978). Evolutionsstrategien. Simulationsmethoden in der medizin und biologie (pp. 83–114). Springer.
- Reeves, C., & Rowe, J. E. (2002). *Genetic algorithms: Principles and perspectives: A guide to ga theory* (Vol. 10). Springer Science & Business Media.
- Reschke, C. H. (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 (Tech. Rep.). 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. Evolutionary computation in economics and finance (pp. 45–60). Springer.
- Robinson, D., & Goforth, D. (2005). *The topology of the 2x2 games: A new periodic table* (Vol. 3). Psychology Press.
- Roca, C. P., Cuesta, J. A., & Sanchez, 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. & Pedrycz, W. (Eds.), Springer handbook of computational intelligence (pp. 825–844). Berlin, Heidelberg: Springer.
- Russell, B. (1959). Common sense and nuclear warfare.
- Savin, I., & Blueschke, D. (2016). Lost in translation: Explicitly solving nonlinear stochastic optimal control problems using the median objective value. *Computational Economics*, 48(2), 317–338.
- Savin, I., Blueschke, D., & Blueschke-Nikolaeva, V. (2018). Slow and steady wins the race: Approximating nash equilibria in nonlinear quadratic tracking games steter tropfen höhlt den stein: Approximation von nash gleichgewichten in nicht-linearen dynamischen spielen. *Jahrbücher für Nationalökonomie und Statistik*, 238(6), 541–569.
- Savin, I., & Egbetokun, A. (2016). Emergence of innovation networks from r & d cooperation with endogenous absorptive capacity. *Journal of Economic Dynamics and Control*, 64, 82–103.
- Schmertmann, C. P. (1996). Functional search in economics using genetic programming. *Computational Economics*, 9(4), 275–298.
- Sefrioui, M., & Perlaux, J. (2000). Nash genetic algorithms: examples and applications. Proceedings of the 2000 congress on evolutionary computation. cec00 (cat. no. 00th8512) (Vol. 1, pp. 509–516).
- Simpson, B. (2004). Social values, subjective transformations, and cooperation in social dilemmas. *Social Psychology Quarterly*, 67(4), 385–395.
- Skyrms, B. (2004). *The stag hunt and the evolution of social structure*. Cambridge University Press.
- Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32(16), 12363–12379.
- Spears, W. M., Jong, K. A. D., Bäck, T., Fogel, D. B., & Garis, H. D. (1993). An overview of evolutionary computation. European conference on machine learning European conference on machine learning (pp. 442–459).
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359.
- Tanimoto, J. (2015). *Fundamentals of evolutionary game theory and its applications*. Springer.
- Von Neumann, J., & Morgenstern, O. (1953). *Theory of games and economic behavior*. Princeton University Press.
- von Stengel, B. (2007). Equilibrium computation for two-player games in strategic and extensive form. *Algorithmic game theory* 53–78.
- Weibull, J. W. (1997). *Evolutionary game theory*. MIT press.

- Widger, J. , & Grosu, D. (2008). Computing equilibria in bimatrix games by parallel support enumeration. 2008 international symposium on parallel and distributed computing (pp. 250–256).
- Yang, G. (2017). Game theory-inspired evolutionary algorithm for global optimization. *Algorithms*, 10(4), 111.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.