Cold play: Learning across bimatrix games

Lensberg, Terje and Schenk-Hoppé, Klaus R.

Department of Finance, NHH–Norwegian School of Economics, Department of Economics, University of Manchester

10 March 2020

Cold play: Learning across bimatrix games

Terje Lensberg[a] and Klaus Reiner Schenk-Hoppé[a,b]

March 10, 2020

Abstract

We study one-shot play in the set of all bimatrix games by a large population of agents. The agents never see the same game twice, but they can learn 'across games' by developing solution concepts that tell them how to play new games. Each agent's individual solution concept is represented by a computer program, and natural selection is applied to derive stochastically stable solution concepts. Our aim is to develop a theory predicting how experienced agents would play in one-shot games.

Keywords: One-shot games, solution concepts, genetic programming, evolutionary stability.

JEL classification: C63, C73, C90.

[a]Department of Finance, NHH–Norwegian School of Economics, Bergen, Norway.
[b]Department of Economics, School of Social Sciences, University of Manchester, United Kingdom.
E-mail: terje.lensberg@nhh.no; klaus.schenk-hoppe@manchester.ac.uk.

By three methods we may learn wisdom: First, by reflection, which is noblest; second, by imitation, which is easiest; and third by experience, which is the bitterest. — Confucius.

1 Introduction

One-shot games put players in unfamiliar situations. Playing well in such situations is a difficult task. Games with multiple Nash equilibria raise the question of which one, if any, of those equilibria will be played, and there is ample evidence that equilibrium solution concepts fail to predict actual behavior in many games. However, by playing many one-shot games, an agent can learn 'across games' to form, and to gradually improve, a theory of games. A theory that can be used to solve all games in a given class is a *solution concept*. This is a map from games to strategy profiles which determines the agent's action in any player position of any game and her conjecture about all other players' actions in that game.

In this paper, we consider the set of all bimatrix games, i.e., two-person simultaneous-move games where each player has a finite number of pure strategies. There is a large population of agents who learn to use individual solution concepts to play games, and our aim is to find a collection of individual solution concepts that forms a stochastically stable equilibrium (SSE) when applied to one-shot bimatrix games.

To obtain an SSE, we represent the individual solution concepts by computer programs and use a genetic programming algorithm (Koza 1992)[1] to evolve these programs until the population mean behavior remains constant. In our context the algorithm works as follows: Begin with a large population of randomly generated programs whose inputs represent information about bimatrix games, and whose outputs can be interpreted as a decision of how to play a game. Let the programs play lots of random games against random opponents and measure their individual performance in those games. Replace some low performing programs with copies of high performing ones; cross and mutate some of the copies and let

[1] A technique for 'programming computers by means of natural selection', see also the series of handbooks edited by Koza (1992-2003).

the programs play another random set of games. By continuing in this manner across thousands of iterations, the programs become increasingly better at one-shot play until possibly, the process converges to an SSE. Taking the mean across all individual solution concepts for each game, one obtains an aggregate solution concept (ASC) which represents the joint distribution of actions and conjectures for each position in every game.

By injecting a flow of randomly generated programs into the population, the genetic programming algorithm creates a noisy environment. On the one hand, this noise raises the bar for making good decisions as the programs have to cope with a population of opponents, some of which will display unexpected or irrational behavior. On the other hand, it is this noise that will ensure that any equilibrium will indeed be stochastically stable, i.e., be robust against innovations in the sense that any deviations by a small number of agents from their current solution concepts would make those agents worse off relative to the remaining population. In other words, an SSE could survive as a real world phenomenon.

Our paper belongs to the literature on learning across games. Following Selten, Abbink, Buchta & Sadrieh (2003), we consider a population of (artificial) agents who use behavior rules as in Stahl (1996) to decide upon some course of action in unfamiliar situations as described by Gilboa & Schmeidler (1995).

Gilboa & Schmeidler (1995) provide a theoretical basis for learning across games. In their 'Case-based decision theory', the agents do not know all states of the world, but they can make decisions by drawing upon their experience with past cases. This situation is what our model is meant to represent. Gilboa, Schmeidler & Wakker (2002) suggest a set of axioms for rational behavior in such situations and show that it can be represented by a similarity-weighted utility function. LiCalzi (1995), Jehiel (2005) and Steiner & Stewart (2008) model learning across games by agents who use exogenous similarity measures, and in Samuelson (2001) and Mengel (2012) the agents learn to partition games into endogeneous analogy classes. An empirical test of Mengel's (2012) partition model is provided by Grimm & Mengel (2012).

Stahl (1996, 1999, 2000) introduced a rule–based approach to model learning by boundedly rational agents. The agents have behavior rules, which are maps from information sets to sets of feasible actions, and the reinforcement principle defines a learning dynamic on the space of behavior rules. In our paper, we use a different learning dynamic, but our solution concepts represent the same idea as Stahl's behavior rules. Stahl's rule based learning model covers a number of special cases, including fictitious play (Brown 1951), replicator dynamics (Taylor & Jonker 1978), belief updating (Mookherjee & Sopher 1994) and reinforcement learning (Roth & Erev 1995). Models of these types have been used by LiCalzi (1995), Germano (2007) and Mengel (2012) to represent learning in theoretical analogy-based models, and by Gale, Binmore & Samuelson (1995), Cooper & Kagel (2003, 2008) and Haruvy & Stahl (2012) to study transfer of learning across games. The latter three papers find that human subjects learn to reason across dissimilar games, and with increasing sophistication as they become more experienced.

Stahl's rule based learning model builds on Nagel (1995) and Stahl & Wilson (1994), who introduced level-k reasoning as a model of initial play. In experiments with initial play, one finds that the subjects often deviate in systematic ways from equilibrium play, and that level-k reasoning and other structural non-equilibrium models (Stahl 2001, Costa-Gomes, Crawford & Broseta 2001) do a better job of predicting actual outcomes. A survey of this literature is provided by Crawford, Costa-Gomes & Iriberri (2013), and a recent contribution is Fudenberg & Liang (2019), who use neural networks to re-examine the empirical evidence. Our paper is related to this literature by considering only one-shot games, but differs in one important respect: In experiments with initial play, the subjects usually play a sequence of one-shot games without intermediate feedback. The purpose is to suppress learning and preserve an impression of initial play throughout the experiment. As a result, inexperienced subjects remain so during the whole experiment. This contrasts with our paper, and with Selten et al. (2003), where the agents receive systematic feedback to become experienced at one-shot play over time.

Selten et al. (2003) is closely related to our paper. They provide a detailed account of an experiment aimed at studying one-shot play in 3×3 games by means of Selten's (1967) *strategy method*. As part of an economics course, students were asked to write computer programs that would determine their choice of actions in randomly chosen 3×3 games. Several contests were held during the teaching term. In each contest the programs played 500,000 random games, with the results of each contest being used by the students to further refine their programs. They quickly introduced a distinction between games with and without pure Nash equilibria. In the former, they ended up coordinating on equilibria with maximal joint payoff. In the latter, their behavior was a more diverse mix of best-reply cascades, as in level-k reasoning.

Also closely related to our paper is a small literature on learning across games by artificial agents. Sgroi & Zizzo (2009) train neural networks (NNs) to play Nash in 3×3 games with one pure equilibrium. They find that the NNs behave as if they try to identify pure Nash equilibria by means of level-k reasoning. When the NNs are applied to unfamiliar games, this 'shortcut' yields a prediction accuracy which is comparable to that of human subjects. Spiliopoulos (2015) considers a population of NNs who learn to play ex post best reply against the field in seven strategically different classes of 2×2 games. He finds strong evidence of cross–game learning, e.g., training on games with more incentives to cooperate yields more cooperation in unfamilar games. Spiliopoulos (2011) uses a population of NNs to play general 3×3 games. He finds that the NNs develop similarity measures which they use to classify games by their strategic properties, consistent with the case-based decision theory of Gilboa & Schmeidler (1995). The same phenomenon occurred in Selten et al.'s (2003) experiment, as mentioned above, and we show that it also occurs in our model.[2]

Many authors have used genetic algorithms to model learning in repeated games and markets. A pioneering contribution to this literature is Arifovic's (1994) analysis of the cobweb market model. Marks (2002) provides a survey, and more recent applications include

[2]See Section 3.3 where the structural properties of solution concepts are analyzed.

coordination games (Chen, Duffy & Yeh 2005), Traveler's dilemma games (Pace 2009), and financial market microstructure models (Lensberg, Schenk-Hoppé & Ladley 2015).

Genetic algorithms impose very little structure on the agents' decision rules. This makes them well suited to model learning in populations of heterogeneous agents. Agents are modeled by specifying their information, their feasible actions and a measure of their individual performance. Competition drives behavior, which is commonly found to agree well with that of human subjects, see e.g, Arifovic (1995, 1996) and Chen et al. (2005).

To our knowledge, our paper is the first to use a genetic algorithm to study learning across games. We show here that a population of agents can learn across one-shot games to solve all finite two-person games. To obtain this result, we impose some structure on the agents' solution concepts to make their task a manageable one. The key element is a separability condition which will allow the agents' programs to process games with different dimensions in the same way. We will also take some steps to enforce a certain degree of rationality. For instance, we will require invariance with regards to the ordering of strategies and invariance with respect to positive affine transformations of payoffs. In addition, to test the robustness of our main result, we will investigate the behavioral consequences of requiring that the agents use rationalizable[3] solution concepts.

Our main result is a new solution concept for one-shot bimatrix games. We examine its logic and performance in detail, and we compare its solutions to many well-known games with the theoretical and empirical evidence. For example, our ASC selects the 'right' solution to Traveler's dilemma games, predicts that the responder will get 40% of the pie in ultimatum games and selects the risk dominant Nash equilibrium in 2×2 games with strict Nash equilibria.

The remainder of the paper is organized as follows: Section 2 describes the model, Section 3 presents the results, and Section 4 concludes.

[3] Bernheim (1984) and Pearce (1984).

2 Model

In this section, we introduce a general class of solution concepts and a genetic programming (GP) algorithm to model their evolution. The algorithm uses a large population of agents, each one equipped with a solution concept that she uses to solve games. Agents will be randomly assigned to play random bimatrix games in some random position, Row (1) or Col (2), against random opponents. By doing 100 independent runs with the GP algorithm, we obtain a detailed data set that can be analyzed to reveal the structure of the evolved solution concepts.

2.1 Solution concepts

Let Γ denote the set of all bimatrix games. The members of Γ are pairs $G = (S, \pi)$, where $S = S_1 \times S_2$ is a finite set of pure strategy profiles and $\pi : S \to \mathbb{R}^2$ is a payoff function such that $\pi(\mathbf{s}) = (\pi_1(\mathbf{s}), \pi_2(\mathbf{s}))$ are the von Neumann-Morgenstern utilities obtained by the two players when profile $\mathbf{s} \in S$ is played. From now on, the word 'game' will be used to designate the members of Γ.

For any game G, let $\Sigma(G)$ denote the associated set of strategy profiles. A *solution concept* is a map F from games to strategy profiles, such that $F(G) \subset \Sigma(G)$ for all $G \in \Gamma$. $F(G)$ can contain one or more elements, any one of which is a solution to G. Solution concepts allow to solve a game from the perspectives of both players (Row and Col). Let $G = (S, \pi)$ be any game and define its transpose $G^\top$ as $G^\top = (S', \pi')$, where $S'_1 = S_2$; $S'_2 = S_1$, and $(\pi'_1(t, s), \pi'_2(t, s)) = (\pi_2(s, t), \pi_1(s, t))$ for all $(s, t) \in S$. Then:

1. each $(s, t) \in F(G)$ is a solution to G from Row's point of view. s is Row's action and t is her conjecture about Col's action; and

2. each $(t', s') \in F(G^\top)$ is a solution to G from Col's point of view. t' is Col's action and s' is his conjecture about Row's action.

One has consistency of actions and conjectures if the solution concept solves any game G at (s,t) if and only if it solves its transpose at (t,s). Nash equilibrium is a solution concept which satisfies this property. In one-shot games one would not expect such consistency to come about because the agents never get a chance to react to false conjectures.

Solution concepts are applied as follows.

Playing games. Let a and b be two agents, equipped with solution concepts F^a and F^b, respectively. Let G be a game and suppose a and b are assigned as player 1 and 2, respectively. The game G is played as follows: Agent a makes a uniform random draw of (s,t) from $F^a(G)$ and plays s. Agent b makes a uniform random draw of (t',s') from $F^b(G^\top)$ and plays t'. a receives payoff $\pi_1(s,t')$ and b receives payoff $\pi_2(s,t')$.

Aggregate solution concepts. Consider a population A of agents, each of whom is equipped with an individual solution concept F^a. For any finite set X, let $|X|$ denote the number of elements in X. For any game G, define

$$p_1^a(s,t,G) \;:=\; \frac{1}{|F^a(G)|} \text{ if } (s,t)\in F^a(G) \text{ and 0 otherwise} \tag{1}$$

$$p_2^a(s,t,G) \;:=\; p_1^a(t,s,G^\top) = \frac{1}{|F^a(G^\top)|} \text{ if } (t,s)\in F^a(G^\top) \text{ and 0 otherwise.} \tag{2}$$

$p_1^a(s,t,G)$ is the probability by which agent a solves G at (s,t) as player 1 (Row) and $p_2^a(s,t,G)$ is the probability by which he solves the transposed game $G^\top$ at (t,s) as player 2 (Col). By taking the mean of the probability distributions $\{(p_1^a,p_2^a)\}_{a\in A}$ across all agents we obtain

$$P_i(s,t,G) \;=\; \frac{1}{|A|}\sum_{a\in A} p_i^a(s,t,G) \tag{3}$$

for each position $i\in\{1,2\}$. $P_1(s,t,G)$ is the percentage of Row players who solve G at (s,t), and $P_2(s,t,G)$ is the percentage of Col players who solve the transposed game $G^\top$ at (t,s). Let $P(s,t,G)=(P_1(s,t,G),P_2(s,t,G))$. The bimatrix $P(\cdot,\cdot,G)$ is the *aggregate solution* to game G for population A, and the function $P(\cdot)$ is the *aggregate solution concept.*

Given an aggregate solution concept P and a game G, one obtains mixed actions and conjectures for the row and column players as the marginal distributions of P, as shown in Table 1.

Table 1: Mixed actions (σ) and conjectures (ϕ) in a game G

$\sigma_1(s,G) := \sum_t P_1(s,t,G)$	Percentage of Row players who do s
$\phi_1(t,G) := \sum_s P_1(s,t,G)$	Percentage of Row players who conjecture that Col will do t
$\sigma_2(t,G) := \sum_s P_2(s,t,G)$	Percentage of Col players who do t
$\phi_2(s,G) := \sum_t P_2(s,t,G)$	Percentage of Col players who conjecture that Row will do s

Mixed Nash equilibria. In our model, the agents solve games by choosing a pair of action and conjecture, using uniform randomizations to select one outcome in games with multiple solutions. There is no mechanism to align the actions or conjectures of indifferent agents to sustain mixed Nash equilibria, which may seem to rig the model in disfavor of such equilibria. However, mixing will also occur at the population level because different agents will typically use (slightly) different solution concepts, and this will enable the population to play mixed Nash equilibria without external intervention. In Section 3.1, we shall see that the agents come very close to playing plausible mixed Nash equilibria in many games.

Numerical representations of solution concepts. A solution concept is (numerically) *representable* if there is a family of functions $V(\cdot,G):\Sigma(G)\to\mathbb{R}$, such that for each game G, $F(G)=\mathrm{argmax}_{\mathbf{s}\in\Sigma(G)}V(\mathbf{s},G)$.

We consider a class of representable solution concepts that includes Nash equilibrium as a special case. For any game $G=(S,\pi)$, and any strategy profile $\mathbf{s}=(s,t)\in S$, define pairs of vectors $\delta(\mathbf{s})=(\delta_1(\mathbf{s}),\delta_2(\mathbf{s}))$ as

$$\delta_1(\mathbf{s}) \;:=\; (\pi_1(s,t)-\pi_1(s',t))_{s'\in S_1\setminus s} \tag{4}$$

$$\delta_2(\mathbf{s}) \;:=\; (\pi_2(s,t)-\pi_2(s,t'))_{t'\in S_2\setminus t}. \tag{5}$$

The vectors (4) and (5) contain the *deviation losses* that players 1 and 2 would incur by unilateral deviations from s and t to each one of their alternative strategies. Next, let $f : \mathbb{R}^2 \to \mathbb{R}$ and $g : \cup_{n \in \mathbb{N}} \mathbb{R}^n \to \mathbb{R}$ be two functions, where, by definition, g takes a variable number of arguments, and define

$$V(\mathbf{s}, G) := f(g(\delta_1(\mathbf{s})), g(\delta_2(\mathbf{s}))). \tag{6}$$

Several key concepts in game theory can be represented in this fashion:

Nash equilibrium. A numerical representation V^N for the (pure strategy) Nash equilibrium concept F^N can be obtained by setting $f(x, y) = \min(x, y)$ and $g(\delta_i(\mathbf{s})) = \min(0, \delta_i(\mathbf{s}))$. This yields

$$V^N(\mathbf{s}, G) := \min\{\min(0, \delta_1(\mathbf{s})), \min(0, \delta_2(\mathbf{s}))\}. \tag{7}$$

Vectors of non-negative deviation losses represent best replies, and a strategy profile $\mathbf{s}$ is a Nash equilibrium in pure strategies if $V^N(\cdot, G)$ attains its maximal value of 0 at $\mathbf{s}$.

Risk dominance. Another special case of (6) is the risk dominance concept of Harsanyi & Selten (1988) for 2×2 games. This is a refinement of the Nash equilibrium concept for that class of games, where the vectors of deviation losses $\delta_i(\mathbf{s})$ are singletons, and where a risk dominant equilibrium is one that maximizes the product of the two players' deviation losses. To represent this solution concept by (6), let g be the identity function on $\mathbb{R}$; $f(x, y) = x \cdot y$ if $(x, y) \geq 0$, and $f(x, y) = -1$ (or any other negative number) otherwise. Then

$$V^{RD}(\mathbf{s}, G) := \begin{cases} \delta_1(\mathbf{s}) \cdot \delta_2(\mathbf{s}) & \text{if } \delta(\mathbf{s}) \geq 0 \\ -1 & \text{otherwise.} \end{cases} \tag{8}$$

Given a game $G = (S, \pi)$, a strategy profile $\mathbf{s}$ is a risk dominant solution if $V^{RD}(\cdot, G)$ attains its maximum on $\Sigma(G)$ at $\mathbf{s}$ with $V^{RD}(\mathbf{s}, G) \geq 0$. Otherwise G has no pure strategy Nash equilibrium and consequently no risk dominant solution.

Interpretation. Any solution concept that is representable by some version of V in

(6) has three features that are worth noting. First, it can be used to solve games of any finite dimension because the function g can take any number of arguments. Second, $V(\cdot) = f(g(\cdot), g(\cdot))$ is separable with respect to the two vectors of deviation losses (the arguments to g). This suggests to think of g as a measure of the extent to which a strategy for one player is a *good reply* to that of the other, and of f as a device that aggregates two good replies into a *good solution*. Third, by relaxing the Nash equilibrium concept in this way, one can construct solution concepts which potentially use more information about games. In particular, it allows to talk about strategies being almost best replies, and to consider if one solution to a game might be better than another because the former provides weaker incentives to deviate than the latter.

The Nash equilibrium concept has some additional properties that do not follow from (6). The following properties will be imposed on (6) as well.

Scale invariance. We will require all solution concepts F to be invariant with respect to positive affine transformation of payoffs, because payoffs are assumed to be Neumann-Morgenstern utilities. Adding a constant term to some player's payoffs has no effect on F because the functions g in (6) only depend on payoff differences, but the functions f and g must be jointly chosen to eliminate any scale effect as well.

Symmetric good replies. A solution concept has symmetric good replies if it is invariant with respect to the ordering of any player's strategies. The Nash equilibrium concept satisfies this property because g^N is symmetric. We will impose this requirement because it prevents the agents from conditioning their actions on some irrelevant aspects of the game.

Iterative good replies. The Nash good reply function, $g^N(\cdot) = \min(\cdot)$, is separable with respect to any subset of arguments. We will impose separability on all functions g in (6). Any such g can then be computed by an iterative algorithm. It is illustrated in Table 2 where $\mathbf{x}$ is a vector of deviation losses and γ is an *iteration function* to compute $g(\mathbf{x})$. $\mathbf{z}$ is a real vector of scratch memory for the algorithm, whose first element (z_1) is taken to be its return value. Sometimes a scalar $\mathbf{z}$ will suffice, in which case it will be denoted z.

Table 2: Algorithm to compute the function g for a player i at strategy combination $\mathbf{s}$ in a game G by means of an iteration function γ. $\mathbf{x} = (x_1, ..., x_K)$ is a vector of length K containing the deviation losses in $\delta_i(\mathbf{s})$ for G at $\mathbf{s} \in \Sigma(G)$ and $d(k)$ is a dummy variable which is 1 if $k = 1$ and 0 otherwise. $\mathbf{z}$ is a real vector of scratch memory for the algorithm, whose first element (z_1) is taken to be its return value.

Pseudo-code	Comment
$\mathbf{z} = \mathbf{0}$	Initialize memory
For $k = 1$ to K	Loop over deviation losses
$\quad\mathbf{z} \leftarrow \gamma(x_k, \mathbf{z}, d(k), K)$	Update memory
End For	End of loop
$g(\mathbf{x}) = z_1$	Return value

For example, defining the iteration function as $\gamma(x_k, z, d(k), K) := \min(x_k, z)$ one obtains $g(\mathbf{x}) = \min(0, \mathbf{x}) = \min(0, \delta_i(\mathbf{s}))$. This is the good reply function used to obtain the Nash equilibrium concept in (7). To compute general good reply functions, the two additional arguments to γ may be needed. K is the number of deviation losses in $\delta_i(\mathbf{s})$; one less than the number of pure strategies available to player i. For instance the value of K can be used by solution concepts that rely on some kind of average. $d(k)$ is a dummy variable to indicate whether the current iteration k is the first one. This information will allow γ to re-initialize one or more of the memory slots $\mathbf{z}$ at the beginning of the first iteration for solution concepts that need some initial value other than 0.

Solution concepts that satisfy iterative good replies have two important benefits: First, they allow to represent games of different dimensions within the same structure and (low-dimensional) domain, parametrized by the game dimensions. Second, this fact, in conjunction with symmetric good replies, will ensure that the solution concepts behave in a similar way across game dimensions. The latter is a desirable property of any solution concept, and without the former our evolutionary approach to solving games would simply not work.

A solution concept F is called *admissible* if and only if it is representable by (6) and satisfies scale invariance, symmetric good replies and iterative good replies. For any such F the associated pair of functions (f, γ) will be said to represent F.

2.2 Implementation of solution concepts

Let F be an admissible solution concept, let (f, γ) be a numerical representation for F, and let g be the good reply function generated by γ by means of the algorithm in Table 2. To solve games, the functions f and γ, which are specific to each agent, must be implemented as computer programs. Because computing time is going to be an issue, we implement f and γ in machine code,[4] following Nordin (1997). Each program consists of at most 32 machine instructions for the x86-64 processor. The processor has 16 floating point registers, and we use four of those as scratch memory for the programs. For the iteration program γ, the contents of the memory slots (denoted $\mathbf{z}$ in Table 2) are preserved across iterations.

Program instructions specify one or more operators and one or more operands. Operators consist of +, -, /, ×, *maximum, minimum, change sign, absolute value*, variable manipulations *copy*, program-flow instructions, *if, goto*, and relational operators $<$, $>$, $\leq$, $\geq$, $=$, $\neq$. This set of operators allows for conditional arithmetic operations and assignments, as well as conditional jumps.[5] Operands consist of the relevant input variables, the four memory slots, and randomly chosen constants. When a program executes, the memory slots are initialized to 0 and the instructions are performed in order. The output from a program is taken to be the value of the first memory slot after the program has executed.[6]

We next describe how scale invariance and symmetric good replies can be imposed on F by means of a 'nudge'. The basic idea is to scramble any information about games that could lead to a violation of the property in question, thereby stimulating development of functional forms that are insensitive to the scrambled information. To explain this idea in detail, we consider a game $G = (S, \pi)$, and a player position $i \in \{1, 2\}$.

[4]The machine code representation is used for fast execution of programs. In addition, we use a byte code representation to simplify program generation and manipulation; a small compiler to translate byte code to binary machine code, and a byte code disassembler to produce program representations that can be read by humans and analyzed by computer algebra applications.

[5]All jumps are forward jumps to avoid infinite loops.

[6]The agents' programs will sometimes produce $\pm\infty$ or NaN (not a number). The function g will be restricted to return only real numbers to ensure that the arguments to f are real, while f will be allowed to return $\pm\infty$ as well. To this end, any NaN or $\pm\infty$ from g and any NaN from f will be replaced by a random draw from a normal distribution with large standard deviation.

First, we impose symmetric good replies by randomly shuffling the deviation losses in $\delta_j(\mathbf{s})$ before computing $g(\delta_j(\mathbf{s}))$ for each player $j \in \{1, 2\}$ and each strategy profile $\mathbf{s}$. This scrambles the ordering of strategies and removes any possibilities for the agents to coordinate, or otherwise condition, their actions on the ordering of strategies.

Second, to impose scale invariance, we introduce a distinction between the payoffs that will be used as arguments to the solution concept F and the payoffs that will be used to measure its performance. To measure performance, we use the original payoffs π_i, whereas the arguments to F are obtained by multiplying both players' payoffs by two separate real random numbers from the interval $[0.01, 100]$. This scrambles the agents' information about the stakes of the game, which provides them with an incentive to develop scale invariant solution concepts.[7]

2.3 Games

Agents develop solution concepts by playing lots of random games. To generate the dimensions and payoffs of those games, a probability distribution on the space of games is needed.

Game payoffs are generated by independent draws from a normal distribution with mean 0 and standard deviation 10. Each payoff is rounded to the nearest integer to produce some games with weak best replies, weakly dominated strategies, and connected components of Nash equilibria. Games with these features are the subject matter of the large literature on equilibrium refinements, and it will be of interest to see if the agents can learn to play such games.

To generate game dimensions, we need a probability distribution with finite support to ensure that the computing time to solve a random game is bounded, and it should select larger games with lower probability in order to save computing time. Moreover, because we shall compare results with alternative experiments where the agents are not allowed to play

[7] As noted earlier, F is already immune against the constant term in such transformations because it only depends on the players' deviation losses. So there is no need to also add a random number.

strictly dominated strategies, we want the game dimensions to be identically distributed across those experiments.

Table 3: Auxiliary probability distribution to select a number n of strategies for one player.

n	2	3	4	5	6	7	8	9	10
$p(n)$	0.222	0.243	0.152	0.117	0.088	0.065	0.050	0.039	0.024

To meet those ends, we consider games where the number of strategies per player is a number between 2 and 10, inclusive. To produce a game G, we first generate a pair of dimensions (n'_1, n'_2) by means of two independent draws from the probability distribution p in Table 3, and then randomly generate payoffs for a game G^1 with those dimensions. Second, we iteratively eliminate all strictly dominated strategies from G^1 to obtain a game G^2 of dimension $(n_1, n_2) \leq (n'_1, n'_2)$. If $n_i < 2$ for any $i \in \{1, 2\}$, we discard G^1 and G^2 and repeat the first two steps until both players in G^2 have at least two undominated strategies. Third, set $G = G^2$ if we want a game without strictly dominated strategies, otherwise, randomly generate a new game G^3 with the same dimensions (n_1, n_2) as G^2, and set $G = G^3$.

The resulting probability distribution on game dimensions selects e.g., 2×2 games with probability 0.21, 4×5 games with probability 0.05 and 10×10 games with probability 0.003.

2.4 Evolution

We apply a genetic programming algorithm (Koza 1992) to model the evolution of solution concepts. The algorithm starts by creating 1,000 random games and 2,000 agents, each equipped with a random pair of programs (f^a, γ^a). These programs are then applied to solve each game for each agent from the point of view of each player, as described in Section 2.1.

The genetic programming algorithm is run for 100,000 iterations, each of which consists of the following three stages:

1. *Performance measurement:* Each agent a plays each game in a random position (1 or 2) against a random opponent $b \neq a$ in the opposite position. The payoffs for player a

are summed up across all games to obtain a measure of a's performance.[8]

2. *Tournament selection:* Using these performance measures, the algorithm arranges 50 tournaments, each involving four randomly selected agents. In each tournament, the algorithm replaces the programs of the two losers by recombining the programs of the two winners. Equipped with new programs, both losers then solve all 1,000 games.

3. *Game replacement:* 10 games are randomly selected and replaced with another 10 randomly generated games. The 10 new games are solved by all 2,000 agents.

By replacing only 10 out of the 1,000 games in stage 3 of each iteration, most games will be played several times by most agents across subsequent iterations. By keeping records of each agent's solutions to each game, it can be solved once and then played repeatedly without having to execute the agent's programs. This allows to complete a run with the genetic programming algorithm in a couple of days, as compared to months if one were to replace all games in every iteration.

With all this repeated play, the reader may wonder what became of our story of one-shot games, in which the agents are supposed to never play the same game twice. Fortunately, it is still intact, because the agents have no memory of previously played games, except for whatever is contained in their programs. From the agents' perspective, the situation looks like a one-shot game, provided the set of games exhibits enough variation over time to prevent overfitting (knowing the solutions to specific games) and induce learning (knowing how to play games). To that end, it will suffice to replace 10 out of 1,000 games in each iteration.

Tournament selection uses the standard genetic operators copy, crossover and mutation to produce programs that perform increasingly better over time. We implement this mechanism as follows:

[8] The performance of a's opponents is computed separately but in the same way, i.e., by randomly selecting an opponent and a position for each game, and accumulating payoffs across all games.

1. *Tournament:* Randomly select four agents from the player population, and rank them by decreasing performance to get an ordered set $\{a_1, a_2, a_3, a_4\}$ of agents.

2. *Copy:* Replace the programs of agents 3 and 4 with copies of the programs of agents 1 and 2. Denote the copied programs by (f^3, γ^3) and (f^4, γ^4).

3. *Crossover:* With probability χ_1, cross f^3 with f^4 by swapping randomly selected sub-lists of instructions among them, and cross γ^3 with γ^4 in the same way.

4. *Mutation:* Each of the four new programs undergoes a mutation with probability χ_2: A single instruction in the program is randomly selected, and replaced with a randomly generated instruction.

The crossover and mutation rates, χ_1 and χ_2, are initially set to 0.5 and 0.8. Between iteration 40,000 and 80,000 both rates decay to 0.01 and stay there until the last iteration. To begin with, this produces a noisy environment with lots of experimentation, and then a period with increasing imitation as the system cools down to possibly settle in a stable state. By collecting data from the last 20,000 iterations, we will examine whether the distribution of solution concepts has then reached a *stochastically stable equilibrium* in the sense of Young (1994).

3 Results

In this section, we present results for the aggregate solution concept (ASC) obtained from the model described in Section 2. We begin by recapitulating a few key details regarding the construction and interpretation of the ASC.

Recall that an agent's behavior in a specific game is determined by her individual solution concept. This is a map from games to strategy profiles which assigns a pair of action and conjecture to each game, conditional on the agent's player position (Row or Col) in the game. An individual solution concept is represented as a pair of programs (f, γ), defined in Section 2.1, where γ is an iteration function to compute a good reply, and f is a good solution function.

To obtain the ASC, we do 100 independent runs with the model. Each run is carried out as described in Section 2.4 with a population of 2,000 agents. At the end of each run, we save the pair of programs (f^a, γ^a) for each agent a. The ASC consists of this collection of 200,000 program pairs. To find the aggregate solution to a given game, we solve it by means of each program pair of the ASC and take the mean of those solutions.

As explained in Section 2.1, the aggregate solution to a given game is a pair $P = (P_1, P_2)$ of probability distributions on the set of strategy profiles for that game, one probability distribution for each of the two players. For a given player i and strategy profile (s, t), $P_i(s, t)$ is the probability that a randomly chosen agent will solve the game at (s, t) when called upon to play it as player i. A Row player does action s, conjecturing that Col will do t, and a Col player does action t, conjecturing that Row will do s. For each probability distribution P_i one derives the mixed actions and conjectures for player i as the marginal distributions of P_i.

It should be noted that actions constitute hard information in the sense of determining the agents' payoffs. Conjectures have no such material basis in one-shot games as there is no way in which the agents can verify their conjectures. But conjectures may still be meaningful as an aid to understanding the agents' decision processes.

Section 3.1 illustrates the behavior of the ASC in some familiar games. Section 3.2 tests convergence and analyzes the performance of the ASC against agents who play best reply, i.e., hypothetical, omniscient agents who know the distribution of strategies in the population for each game. We also look at the performance of the ASC against Nash players in games with one pure Nash equilibrium. Section 3.3 looks into the structure of individual and aggregate solution concepts by investigating the functional form of the good reply iteration function γ and the good solution function f. The aim is to understand the logic that drives the aggregate behavior. In Section 3.4 we carry out two robustness checks. In the first one, we test whether the agents' good reply iteration functions are sensitive to initial conditions, and in the second one, we investigate the consequences of imposing rationalizability upon the agents' solution concepts.

3.1 Behavior in selected games

When assessing the results of this section, it is important to bear in mind that the agents have no prior experience with any of the games to be considered here. Anything the agents do has been learned by experience with other games, and so the situation is literally one-shot play by experienced agents.

3.1.1 Classical games

Battle of the sexes. In this game, Row and Col would like to attend a Ballet or a Football match. Row prefers Ballet, Col prefers Football, but in any case, they would like to be together. Panel (a) of Table 4 contains the payoff matrix, with the two pure strategy Nash equilibria indicated in boldface. There is a third (mixed) equilibrium, where both agents play their preferred action with 60% probability.

Table 4: Battle of the sexes. Numbers in italics are probabilities (%).

(a) Payoffs, actions and conjectures

(s,t)		B	F	
	σ	*41*	*59*	Col
B	*59*	**3, 2**	0, 0	*41*
F	*41*	0, 0	**2, 3**	*59*
Row		*59*	*41*	ϕ

(b) Solution, actions and conjectures

(s,t)		B	F	
	σ	*41*	*59*	Col
B	*59*	*59*,*41*	*0*, *0*	*41*
F	*41*	*0*, *0*	*41*,*59*	*59*
Row		*59*	*41*	ϕ

Panel (b) of Table 4 shows the aggregate solution $P = (P_1, P_2)$ and its marginal distributions $\sigma = (\sigma_1, \sigma_2)$ and $\phi = (\phi_1, \phi_2)$. The marginals (σ_1, σ_2) are the aggregate mixed actions of the Row and Col players, and the marginals (ϕ_1, ϕ_2) are their aggregate conjectures about the opponent's actions. The mixed actions and conjectures are also shown along with the payoff matrix in Panel (a).

Panel (b) shows that 59% of both players solve the game at their preferred Nash equilibrium, which is (B, B) for Row and (F, F) for Col, while 41% solve it at the one preferred by the other player. There are several things to note about this solution. First, it yields the mixed actions $0.59B + 0.41F$ for Row, and $0.41B + 0.59F$ for Col, which almost exactly match the mixed Nash equilibrium of the game. Second, since the players solve the game at (B, B) and (F, F) with different probabilities, their solutions cannot result from individual uniform randomizations between equally good solutions. To obtain the solution in Panel (b), there must be some mixing at the population level. Third, the agents' conjectures are wrong: The Row players do B and F with probabilities 59 and 41% while the Col players believe they do it with the opposite probabilities. However, these inconsistencies could persist in repeated play because the mixed actions are almost a Nash equilibrium.

Rock, Paper, Scissors. This zero-sum game, depicted in Table 5, has a unique (mixed) Nash equilibrium in which both players play each of their three actions with probability 1/3. The ASC yields the same actions, and (correct) conjectures.

Table 5: Rock, Paper, Scissors. Numbers in italics are probabilities (%).

(a) Payoffs, actions and conjectures

(s,t)		R	P	S	
	σ	*33*	*33*	*33*	Col
R	*34*	0, 0	-1, 1	1, -1	*33*
P	*33*	1, -1	0, 0	-1, 1	*33*
S	*33*	-1, 1	1, -1	0, 0	*33*
	Row	*33*	*33*	*33*	ϕ

(b) Solution, actions and conjectures

(s,t)		R	P	S	
	σ	*33*	*33*	*33*	Col
R	*34*	*19,19*	*1,13*	*13, 1*	*33*
P	*33*	*13, 1*	*19,19*	*1,13*	*33*
S	*33*	*1,13*	*13, 1*	*19,19*	*33*
	Row	*33*	*33*	*33*	ϕ

Consider next the details of the solution shown in Panel (b). Given the payoff structure of this game, it seems fair to say that $3 \times 19 = 57\%$ of both players believe in a draw; $3 \times 13 = 39\%$ expect to win, and $3 \times 1 = 3\%$ expect to lose. On the other hand, the agents' tendency to solve the game at the diagonal suggests that they may rather be looking for some kind of equitable compromise. With that interpretation in mind, the agents appear to be 57% egalitarian, 39% selfish, and 3% altruistic.

Prisoners' dilemma. We next consider a game where the agents' self-interest prevails.

In 'Prisoners' dilemma', Table 6, the players get a sentence depending on whether they deny (d) or confess (C) a crime. Deny is strictly dominated[9], (C, C) is the only Nash equilibrium, and this solution is also selected by 100% of the agents, so (C, C) is the ASC outcome.

Table 6: Prisoners' dilemma. Numbers in italics are probabilities (%).

(a) Payoffs, actions and conjectures

(s,t)		d	C	
	σ	*0*	*100*	Col
d	*0*	-1, -1	-3, 0	*0*
C	*100*	0, -3	**-2**, **-2**	*100*
	Row	*0*	*100*	ϕ

(b) Solution, actions and conjectures

(s,t)		d	C	
	σ	*0*	*100*	Col
d	*0*	*0*,*0*	*0*, *0*	*0*
C	*100*	*0*,*0*	*100*,*100*	*100*
	Row	*0*	*100*	ϕ

In games where (almost) all agents agree on one strategy profile, the solution bimatrix in Panel (b) is not informative and will not be shown from now on.

3.1.2 Refinements

We continue with some games from the refinement literature, which analyzes strategic stability of Nash equilibria with respect to criteria such as subgame perfectness, weak dominance, and backward and forward induction. The question is whether, or to what extent, the ASC reflects such considerations.

Table 7: Market entry game. Numbers in italics are probabilities (%).

(s,t)		F	A	
	σ	*0*	*100*	Col
O	*3*	**2**, **2**	2, 2	*0*
E	*97*	0, 0	**3**, **1**	*100*
	Row	*3*	*97*	ϕ

Market entry game. In this game, which is shown in Table 7, Col is an incumbent monopolist. Row can stay out of the market (O) or enter (E), in which case Col can choose to fight (F) or acquiesce (A). The game has two Nash equilibria in pure strategies, indicated

[9] We use lower case letters to designate actions that do not survive iterated elimination of strictly dominated actions.

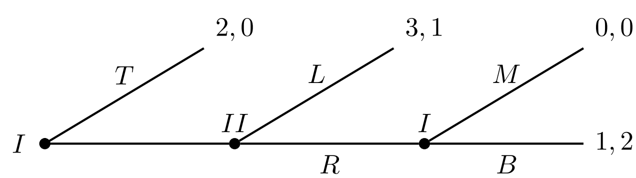
by bold type. Backward induction supports (E, A), and so does the ASC, which plays this pair of strategies with 97% probability.

The next two games are taken from Kohlberg & Mertens (1986).

Kohlberg and Mertens I. The game in Table 8 has two pure Nash equilibria, (T, R) and (M, L), and a unique strategically stable set, which is the convex hull of (T, R) and $(T, \frac{1}{2}L + \frac{1}{2}R)$. Backward induction selects (T, R) with payoffs $(2, 0)$, but (M, L) is supported by the following (informal) forward induction argument: If Row fails to play T, then Col should understand that Row aims to get 3 by threatening to play M if Col fails to play L. This yields (M, L), which is the solution selected by the ASC.

Table 8: Kohlberg & Mertens (1986, p. 1029). Numbers in italics are probabilities (%).

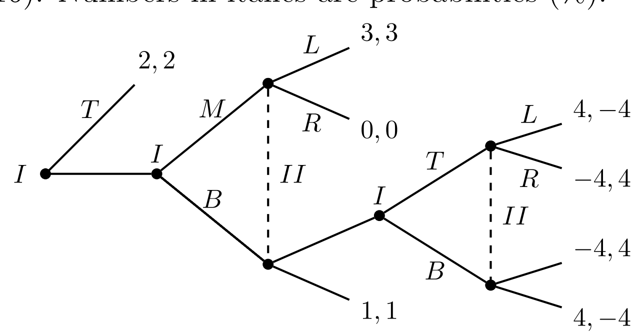
(s, t)		L	R	
	σ	*99*	*1*	Col
T	*4*	$2, 0$	$\mathbf{2, 0}$	*0*
M	*95*	$\mathbf{3, 1}$	$0, 0$	*99*
B	*0*	$3, 1$	$1, 2$	*1*
	Row	*95*	*5*	ϕ



Kohlberg and Mertens II. The game in Table 9 has one Nash equilibrium in pure strategies (T, R) with payoffs $(2, 2)$, and a mixed equilibrium $(M, \frac{1}{2}LL + \frac{1}{2}LR)$ with superior payoffs $(3, 3)$, which is selected by the ASC. By replacing the subgame with its value (0) and applying iterated dominance, one finds that the mixed equilibrium is also the unique strategically stable set of this game.

Table 9: Kohlberg & Mertens (1986, p. 1016). Numbers in italics are probabilities (%).

(s, t)		LL	LR	R	
	σ	*50*	*50*	*0*	Col
T	*0*	2, 2	2, 2	$\mathbf{2, 2}$	*0*
M	*100*	3, 3	3, 3	$0, 0$	*100*
BT	*0*	4, -4	-4, 4	$1, 1$	*0*
BB	*0*	-4, 4	4, -4	$1, 1$	*0*
	Row	*50*	*50*	*0*	ϕ



3.1.3 Equilibrium selection

We next apply the aggregate solution concept to some games in which refinement considerations somehow fail to identify the 'right' outcome with respect to intuition or empirical evidence.

Stag hunt. This game, which is due to Carlson & van Damme (1993), represents the following story: Two hunters can cooperate (C) to catch a stag, or hunt alone (A) to obtain a catch of smaller game amounting to a fraction $x \in (0,1)$ of what each of them would get by cooperating.

Table 10: Stag hunt game. Numbers in italics are probabilities (%).

(a) $x < \frac{1}{2}$

(s,t)		C	A	
	σ	*100*	*0*	Col
C	*100*	**1, 1**	0, x	*100*
A	*0*	x, 0	**x, x**	*0*
	Row	*100*	*0*	ϕ

(b) $x > \frac{1}{2}$

(s,t)		C	A	
	σ	*0*	*100*	Col
C	*0*	**1, 1**	0, x	*0*
A	*100*	x, 0	**x, x**	*100*
	Row	*0*	*100*	ϕ

The game is illustrated in Table 10. It has two strict Nash equilibria: (C, C) and (A, A). When $x < \frac{1}{2}$, the Risk Dominant equilibrium (Harsanyi & Selten 1988) is (C, C), and when $x > \frac{1}{2}$, it is (A, A). Table 10 shows that the ASC always selects the risk dominant equilibrium in the Stag hunt game. When $x = \frac{1}{2}$ (not shown in the table), 50% of the agent population solve the game at (C, C) and 50% solve it at (A, A).

Ultimatum game. Few games have been subject to more empirical analysis than the Ultimatum game of Güth, Schmittberger & Schwarze (1982). In this game, Row and Col get n dollars to share if they can agree how to do it. Row (the proposer) suggests a division by offering an integer amount of x dollars to Col (the responder). Col accepts or rejects. If he accepts, they divide according to Row's suggestion, if Col rejects the offer, both get zero. Any division of the money is the outcome of some Nash equilibrium, but only one is subgame perfect: Row offers zero dollars and Col accepts any offer.

A small version of this game (with 5 dollars to share) is shown in Table 11. Action O_k

Table 11: Ultimatum game. Numbers in italics are probabilities (%).

(s,t)	σ	A_0 *0*	A_1 *0*	A_2 *100*	A_3 *0*	A_4 *0*	A_5 *0*	Col
O_0	0	$\mathbf{5},\mathbf{0}$	$0,0$	$0,0$	$0,0$	$0,0$	$\mathbf{0},\mathbf{0}$	*0*
O_1	*0*	$4,1$	$\mathbf{4},\mathbf{1}$	$0,0$	$0,0$	$0,0$	$0,0$	*0*
O_2	*100*	$3,2$	$3,2$	$\mathbf{3},\mathbf{2}$	$0,0$	$0,0$	$0,0$	*100*
O_3	*0*	$2,3$	$2,3$	$2,3$	$\mathbf{2},\mathbf{3}$	$0,0$	$0,0$	*0*
O_4	*0*	$1,4$	$1,4$	$1,4$	$1,4$	$\mathbf{1},\mathbf{4}$	$0,0$	*0*
O_5	*0*	$0,5$	$0,5$	$0,5$	$0,5$	$0,5$	$\mathbf{0},\mathbf{5}$	*0*
	Row	*0*	*0*	*100*	*0*	*0*	*0*	ϕ

for Row stands for 'Offer k dollars', and action A_k for Col stands for 'Accept any offer of k or more dollars'. In the ASC, Row offers 2 dollars, and Col accepts all offers of 2 or more. If the total amount is doubled to 10 from 5 dollars, the ASC offers and demands double to 4 from 2. These results agree well with the experimental evidence, where mean offers amount to some 40% of the stake, and where the responder rejects offers of some 30% or less, see, e.g., Güth & Tietz (1990).

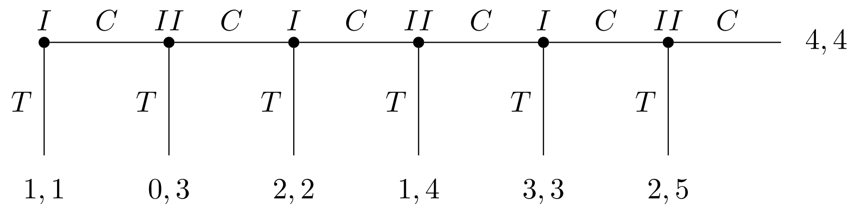
3.1.4 Non-equilibrium behavior

The ultimatum game challenges the idea of backward induction – a basic rationality postulate in game theory. We next consider some games where intuition or experiment suggest that the players will not even play a Nash equilibrium.

The Centipede game by Rosenthal (1981) describes a situation in which two players alternate to decide when to take (T) an increasing pot of money. By continuing (C) for one more round, a player gains if the other player also continues, but loses if the other player then decides to take. A version of this game is shown in Table 12. For each player, C_n denotes the strategy of n C's and then a T if $n < 3$.

The game has a unique (subgame perfect) Nash equilibrium, in which both players take at the first opportunity. In experiments with human subjects, the game often continues for several moves, but very seldom to the end (McKelvey & Palfrey 1992). Under the ASC, 77%

Table 12: Centipede game. Numbers in italics are probabilities (%).



(a) Payoffs, actions and conjectures

(s,t)		C_0	C_1	C_2	C_3	
	σ	*8*	*4*	*67*	*21*	Col
C_0	*22*	**1, 1**	1, 1	1, 1	1, 1	*9*
C_1	*1*	0, 3	2, 2	2, 2	2, 2	*3*
C_2	*0*	0, 3	1, 4	3, 3	3, 3	*2*
C_3	*77*	0, 3	1, 4	2, 5	4, 4	*86*
	Row	*20*	*1*	*19*	*60*	ϕ

(b) Solution, actions and conjectures

(s,t)		C_0	C_1	C_2	C_3	
	σ	*8*	*4*	*67*	*21*	Col
C_0	*22*	*20,6*	*1,1*	*1, 1*	*1, 1*	*9*
C_1	*1*	*0,2*	*0,0*	*1, 1*	*0, 0*	*3*
C_2	*0*	*0,0*	*0,2*	*0, 0*	*0, 0*	*2*
C_3	*77*	*0,0*	*0,1*	*17,65*	*60,20*	*86*
	Row	*20*	*1*	*19*	*60*	ϕ

of the Row players continue as long as they can, and 86% of the Col players conjecture they will do so. However, Row's willingness to continue seems to be based on the false conjecture that 60% of the Col players will also continue until the end, whereas only 21% of them actually plan to do so. The mixed actions for this game imply that 22% of the player pairs end the game at the first opportunity with payoffs $(1, 1)$; $0.77 \times 0.67 = 52\%$ end it at the next to last node with payoffs $(2, 5)$; and $0.77 \times 0.21 = 16\%$ go all the way to the end with payoffs $(4, 4)$.

Traveler's dilemma. In this game, due to Basu (1994), two travelers have lost their luggage and the airline offers compensation for their loss. They can claim any integer amount in the interval $[\underline{c}, \overline{c}] = [2, 100]$. In any case, the airline will pay both travelers the minimum of the two claims, with the following (slight) modification: If player i claims more than player j, then i pays a penalty of $R = 2$ dollars, and j is rewarded by the same amount. As noted by Basu (1994), intuitively both players should make a high claim and pay little attention to the small penalty/reward. However, the game has a unique Nash equilibrium where both players claim the minimal 2 dollars. In fact, this is the only action pair which survives iterated elimination of strictly dominated strategies.

Capra, Goeree, Gomez & Holt (1999) conduct an experiment with human subjects and find that their behavior is sensitive to the penalty/reward parameter R, with players making large claims for small R and vice versa. The ASC turns out to have the same property. To illustrate, we consider a small version of the Traveler's dilemma game, where $(\underline{c}, \overline{c}) = (4, 11)$ instead of $(2, 100)$. The game is shown in Table 13, where C_n and c_n stand for 'Claim n dollars'.

Table 13: Traveler's dilemma game with $\underline{c} = 4$, $\overline{c} = 11$ and penalty/reward parameter $R = 2$. Numbers in italics are probabilities (%).

(s,t)		C_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	
	σ	*50*	*0*	*0*	*0*	*0*	*0*	*0*	*50*	Col
C_4	*50*	$\mathbf{4,4}$	$6,2$	$6,2$	$6,2$	$6,\ 2$	$6,\ 2$	$6,\ 2$	$6,\ 2$	*50*
c_5	*0*	$2,6$	$5,5$	$7,3$	$7,3$	$7,\ 3$	$7,\ 3$	$7,\ 3$	$7,\ 3$	*0*
c_6	*0*	$2,6$	$3,7$	$6,6$	$8,4$	$8,\ 4$	$8,\ 4$	$8,\ 4$	$8,\ 4$	*0*
c_7	*0*	$2,6$	$3,7$	$4,8$	$7,7$	$9,\ 5$	$9,\ 5$	$9,\ 5$	$9,\ 5$	*0*
c_8	*0*	$2,6$	$3,7$	$4,8$	$5,9$	$8,\ 8$	$10,\ 6$	$10,\ 6$	$10,\ 6$	*0*
c_9	*0*	$2,6$	$3,7$	$4,8$	$5,9$	$6,10$	$9,\ 9$	$11,\ 7$	$11,\ 7$	*0*
c_{10}	*0*	$2,6$	$3,7$	$4,8$	$5,9$	$6,10$	$7,11$	$10,10$	$12,\ 8$	*0*
c_{11}	*50*	$2,6$	$3,7$	$4,8$	$5,9$	$6,10$	$7,11$	$8,12$	$11,11$	*50*
	Row	*50*	*0*	*0*	*0*	*0*	*0*	*0*	*50*	ϕ

When $R = R^* \equiv 2$, the agents make the minimal and maximal claims with equal probability, as shown in Table 13. When $R > R^*$, all agents claim the minimal 4, and when $R < R^*$ all agents claim the maximal 11. The critical value R^*, relative to the length of the feasible claim interval is $R^*/(\overline{c} - \underline{c}) = 2/(11 - 4) = 0.29$, which is in line with the empirical findings of Capra et al. (1999).

Social norms. There is a large literature on the role of social norms in economic transactions and relationships. In experiments with human subjects on bargaining, public goods, and labor relations, the hypothesis of purely self-interested behavior is often rejected in favor of explanations based on fairness, reciprocity or altruism. We have applied our solution concept to some of the games studied in this literature and found that in many cases, the ASC agrees with the empirical results in the sense of predicting more cooperation than what would be achieved through rational play by self-interested agents.

To illustrate, consider the gift exchange experiment of Van der Heijden, Nelissen, Potters & Verbon (1998). Two players live for two periods. A player who consumes c_1 in period 1 and c_2 in period 2 obtains utility $c_1 \cdot c_2$. In period 1, player 1 is rich and player 2 is poor. In period 2 their situations are reversed. A rich player has income 9 and a poor player has income 1, but the players can smooth consumption by exchanging gifts: Player 1 gives an integer amount $0 \leq s \leq 7$ to player 2 in period 1 and player 2 gives $0 \leq t \leq 7$ to player 1 in period 2. This yields utilities

$$
\begin{aligned}
u_1(s,t) &= (9-s)\cdot(1+t) && (9)\\
u_2(s,t) &= (9-t)\cdot(1+s) && (10)
\end{aligned}
$$

for players 1 and 2, respectively. The simultaneous move version of this game is shown in Table 14, where t_k stands for 'Transfer k dollars to the other player'. Giving zero (T_0) strictly dominates any other action for both players, but the ASC predicts that both players will give one dollar (t_1) to the other player. This agrees with the average gifts of 0.99 and 1.03 observed empirically by Van der Heijden et al. (1998).

Table 14: Gift exchange game. Numbers in italics are probabilities (%).

(s,t)		T_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	
	σ	*0*	*100*	*0*	*0*	*0*	*0*	*0*	*0*	Col
T_0	*0*	**9, 9**	18, 8	27, 7	36, 6	45, 5	54, 4	63, 3	72, 2	*0*
t_1	*100*	8, 18	16, 16	24, 14	32, 12	40, 10	48, 8	56, 6	64, 4	*100*
t_2	*0*	7, 27	14, 24	21, 21	28, 18	35, 15	42, 12	49, 9	56, 6	*0*
t_3	*0*	6, 36	12, 32	18, 28	24, 24	30, 20	36, 16	42, 12	48, 8	*0*
t_4	*0*	5, 45	10, 40	15, 35	20, 30	25, 25	30, 20	35, 15	40, 10	*0*
t_5	*0*	4, 54	8, 48	12, 42	16, 36	20, 30	24, 24	28, 18	32, 12	*0*
t_6	*0*	3, 63	6, 56	9, 49	12, 42	15, 35	18, 28	21, 21	24, 14	*0*
t_7	*0*	2, 72	4, 64	6, 56	8, 48	10, 40	12, 32	14, 24	16, 16	*0*
	Row	*0*	*100*	*0*	*0*	*0*	*0*	*0*	*0*	ϕ

To understand why the agents sometimes act as if motivated by social norms, recall that our solution concepts solve games $G = (S, \pi)$ at strategy profiles $\mathbf{s} \in S$ which maximize

$$f(g(\delta_1(\mathbf{s})), g(\delta_2(\mathbf{s}))),$$

where $\delta_1(\mathbf{s})$ and $\delta_2(\mathbf{s})$ are vectors of deviation losses for players 1 and 2. The function $f(g(\cdot), g(\cdot))$ resembles a social welfare function for the two players, except that its arguments are deviation losses instead of payoffs. But in many games, including the Gift exchange game in Table 14 and the Rock, Paper, Scissors game in Table 5, payoffs and deviation losses are positively correlated, so when the ASC solves such games by balancing the players' incentives to deviate, it looks as if it tries to make a fair compromise in terms of payoffs.

3.2 Performance and stability

We have seen that the aggregate solution concept (ASC) sometimes solves games at strategies that do not constitute a Nash equilibrium. In this section we examine how often non-Nash play occurs, how costly it is relative to always playing best reply (if one could) and what non-Nash behavior means in terms of evolutionary stability. We also test whether the 100 model runs have converged to stochastically stable equilibria.

Table 15 contains descriptive statistics for a set of variables that measure the performance and stability of the ASC. The performance variables in Panel 1 are computed for each of the 100 model runs from five equally spaced samples taken from the last 2,000 (out of 100,000) iterations. *Consensus* is the percentage of agents who play the modal strategy for a given game and position. With a value close to 100%, it shows that there is very little intra-run heterogeneity among the agents. *playBestReply* is the percentage of agents whose actions are a best reply to the ASC; *meanPayoff* is the mean payoff of the ASC against the ASC, and *gainBR* is the percentage net gain in mean payoff from playing best reply, rather than ASC, against the ASC. *Consensus*, *playBestReply* and *meanPayoff* are computed separately for each game and each position and then averaged across all games and positions. *gainBR*

Table 15: Descriptive statistics. The number of observations is 100 for each variable, one observation for each of the 100 independent runs of the GP-algorithm.

Variable		Mean	Std.dev	Min	Max
Panel 1: All games					
Consensus		98.8%	0.2%	98.3%	99.8%
playBestReply		83.8%	0.4%	82.5%	84.8%
meanPayoff		8.90	0.09	8.62	9.15
gainBR		8.0%	0.3%	7.4%	8.9%
Panel 2: Games with one pure Nash equilibrium					
gain2BR		-7.8%	0.6%	-9.4%	-6.1%
pASC_ASC	(a)	9.09	0.06	8.96	9.24
pNash_ASC	(b)	7.76	0.07	7.58	7.91
pNash_Nash	(c)	8.69	0.06	8.57	8.86
pASC_Nash	(d)	7.51	0.07	7.34	7.67
pDiff	(a-b) - (c-d)	0.16	0.03	0.09	0.23
playNash		83.8%	0.4%	83.0%	84.6%

is computed at an aggregate level because game payoffs are normally distributed with a zero mean.

The variables in Panel 2 of Table 15 are intended to provide some information about the evolutionary stability of the ASC. Data are obtained by restarting each saved population to solve 10,000 random games with exactly one Nash equilibrium in pure strategies. *gain2BR* is the percentage net gain to player i from deviating to a best reply (if not currently playing a best reply) when that is followed by subsequent best reply by player j; *pASC_ASC* is the mean payoff across all games and positions from playing the ASC against itself (identical to *meanPayoff* in Panel 1 except for considering only games with one pure Nash equilibrium); *pNash_ASC* is the mean payoff from playing the Nash equilibrium actions against the ASC; *pNash_Nash* is the mean payoff from playing the Nash equilibrium against itself; *pASC_Nash* is the mean payoff from playing the ASC against the Nash equilibrium; *pDiff* is the net gain from playing the ASC (rather than Nash) against ASC, minus the net gain from playing Nash (rather than ASC) against Nash, and *playNash* is the joint probability of Row and Col playing the pure Nash equilibrium.

The findings in Table 15 can be interpreted as follows. The ASC appears to be well protected against invasion by agents who play Nash because by switching from ASC to Nash they would lose on average $1.33 = 9.09 - 7.76$ (*pASC_ASC* - *pNash_ASC* in Panel 2). The agents play best reply to the ASC 83.8% of the time, which yields an average payoff of 8.90 (*meanPayoff* in Panel 1). An agent could increase her average payoff by 8% if she could play best reply in every game (*gainBR*), but if every deviation to best reply would trigger another best reply from the opponent, the 8% gain would turn into a 7.8% loss (*gain2BR*). Finally, *pDiff* shows that ASC agents outperform Nash agents in an ASC world by a larger margin than Nash agents outperform ASC agents in a Nash world. In other words, ASC is more robust against invasion by Nash agents than vice versa.

We next perform a simple test to check if the 100 model runs have converged to stochastically stable equilibria. This is done by testing for trends in the four variables in Panel 1 of Table 15 towards the end of the model runs. To that end, we use data sampled at every 500th iteration from the last 20,000 iterations of each model run, when mutation and crossover probabilities have reached their common minimum of 1%. We skip the middle part of the data set and test for differences in means between the two intervals 80,000–85,000 and 95,000–100,000 of iterations. The boundary points of each interval are included, which yields 2×11 observations for each run and 2,200 observations in total for each variable in Table 16. The results are consistent with the hypothesis that the 100 model runs have reached stochastically stable equilibria after 80,000 iterations.

Table 16: Convergence tests. Tests of differences in means for the variables *Consensus*, *playBestReply*, *meanPayoff* and *gainBR* across two intervals of model iterations. The number observations is 2,200 for each variable.

Iterations	*Consensus*	*playBestReply*	*meanPayoff*	*gainBR*
80,000 – 85,000	98.8%	83.8%	8.91	8.0%
95,000 – 100,000	98.8%	83.8%	8.90	8.0%
p-value	(0.634)	(0.883)	(0.118)	(0.796)

3.3 Structural properties of solution concepts

Recall that the individual solution concept for an agent a is represented by a pair of computer programs (f^a, γ^a), where f^a is a good solution function and γ^a is an iterator which is used to compute the agent's good reply function g^a. In this section, we aim to uncover structural properties of these programs to shed light on the results presented above. In addition, we will analyze the aggregate solution concept, which is derived from the collection of all individual programs.

The programs of a typical agent is provided in (11) and (12):[10]

$$\gamma^a(x, z) = z + 0.006 + 4x. \tag{11}$$

$$f^a(x_1, x_2) = \begin{cases} x_1 \cdot x_2, & \text{if } (x_1, x_2) > 0 \\ x_2, & \text{if } x_1 > 0 \text{ and } x_2 \leq 0 \\ -\infty, & \text{if } x_1 < 0 \\ \text{undefined}, & \text{if } x_1 = 0. \end{cases} \tag{12}$$

The good reply score equates to[11]

$$g^a(\mathbf{x}) = \sum_{k=1}^{K} (0.006 + 4x_k). \tag{13}$$

Note that g^a is additive and almost proportional to the sum of deviation losses. Thanks to the constant 0.006 in (11), $g^a(\mathbf{x})$ is positive if $\sum_{k=1}^K x_k \geq 0$ and negative almost always if $\sum_{k=1}^K x_k < 0$. In turn, the function $f^a(g^a(\cdot), g^a(\cdot))$ extends continuously from positive to zero sums of deviation losses and almost never returns undefined values.[12]

[10] The programs have been simplified and the constants are truncated. To simplify a program, first evaluate it on one million data points. Second, for each instruction in the program, tentatively replace it by a NOP (no operation), then re-evaluate the program on each data point. Accept the NOP if the change had no effect on the output, otherwise keep the original instruction. Third, continue in this manner until no further instructions can be replaced by NOPs without affecting output.

[11] Given a K-vector $\mathbf{x}$ of deviation losses, initialize z to 0, iterate $z \leftarrow \gamma(x_k, z)$ for $k = 1, \ldots, K$, and finally set $g^a(\mathbf{x}) = z$ to obtain (13).

[12] To see this, recall that each x_k is a random integer c_k, scaled by some random real $\alpha \in [0.01, 100]$, hence $\sum_{k=1}^K x_k = \alpha \sum_{k=1}^K c_k \in (-\infty, -0.01] \cup \{0\} \cup [0.01, \infty)$. Consequently, $g^a(\mathbf{x}) \neq 0$ almost surely, and

To solve a game, one selects a strategy profile which maximizes $f^a(g^a(\cdot), g^a(\cdot))$. Since $g^a(\mathbf{x}) > 0$ if $\sum_{k=1}^{K} x_k \geq 0$, then, for each conjecture t about player 2, there is an action s for player 1 (e.g., a best reply to t) which yields a positive g^a-score to player 1 at (s, t). This implies that $f^a(g^a(\cdot), g^a(\cdot))$ is maximized at case 1 or 2 of (12). *Case 1:* Games with strategy profiles that yield two positive g^a-scores (e.g., pure Nash equilibria) are solved at some strategy profile (not necessarily a Nash equilibrium) which maximizes their product. *Case 2:* All other games are solved at some strategy profile that maximizes the (non-positive) g^a-score to player 2 among those that yield positive g^a-scores to player 1. In other words, the action is a good reply to the conjecture, which is a least bad reply to any such action.

The ASC, which consists of the pairs of programs of 200,000 agents, is a more complex object than an individual agent's solution concept. To study the ASC, we proceed in two steps. First, we describe the behavior of the ASC in 2×2 games. Second, we show that this behavior generalizes in a natural way to larger games because the additive structure of the good reply function in (13) is shared by almost all agents.

3.3.1 2×2 games

Let Γ^2 denote the set of 2×2 games. Consider an agent a and the composite function $v^a : \mathbb{R}^2 \to \mathbb{R}$, defined as

$$v^a(x) := f^a(g^a(x_1), g^a(x_2)), \tag{14}$$

where $x = (x_1, x_2) = \delta(\mathbf{s})$ is a pair of deviation losses corresponding to some strategy profile $\mathbf{s}$ for some $G \in \Gamma^2$. The function v^a in (14) is a numerical representation of agent a's solution concept restricted to Γ^2. To play a game $G = (S, \pi) \in \Gamma^2$, a maximizer $\mathbf{s} = (s, t)$ of $v^a(\delta(\cdot))$ on S is choosen. The agent does s as player 1 and conjectures that the opponent will do t as player 2.

$g^a(\mathbf{x}) \geq 0.006K > 0$ whenever $\sum_{k=1}^{K} x_k \geq 0$. But g^a may fail to preserve the sign of $\sum_{k=1}^{K} x_k$ when $\sum_{k=1}^{K} x_k$ is negative and close to 0. The maximal negative value of $\sum_{k=1}^{K} x_k$ is -0.01. Then $g^a(\mathbf{x}) = 0.006K - 0.040$, which is negative if and only if $K < 6.67$. So $g^a(\mathbf{x})$ preserves the sign of $\sum_{k=1}^{K} x_k$ for $K \leq 6$, but may fail to do so for $K \geq 7$. However, such failures occur less than twice per million random games, which explains why this 'bug' escapes removal by the genetic programming algorithm.

We approximate the ASC on Γ^2 by deriving a numerical representation v. It is constructed as follows: Let D_0 be a finite two-dimensional grid of pairs of deviation losses. For each agent a, rank all the points in D_0 according to v^a, and break ties by randomly ordering the members of each tied set of points. Define $v : D_0 \to [0, 1]$ as the Borda count of all the 200,000 rankings obtained in this way, normalized to yield rank scores in $[0, 1]$. Let D denote the convex hull of D_0 and extend v to D by interpolation. For each $G = (S, \pi) \in \Gamma^2$ such that $\delta(S) \subset D$, let $F^2(G) := \text{argmax}_{\mathbf{s} \in S} v(\delta(\mathbf{s}))$. Then F^2 is a representative solution concept, approximating the ASC for 2×2 games.

A contour plot of v is provided in Figure 1 for $D_0 = \{-10, -9.5, \ldots, 10\}^2$. The four subsets $\{D_k\}_{k=1}^4$ of D are defined as

$$\begin{aligned} D_1 &= \{x \in D \mid (x_1, x_2) > 0\}, & D_3 &= \{x \in D \mid x_1 \geq 0, x_2 < 0\} \cup \{x \in D \mid x_1 = 0, x_2 \geq 0\}, \\ D_2 &= \{x \in D \mid x_1 > 0, x_2 = 0\}, & D_4 &= \{x \in D \mid x_1 < 0\}. \end{aligned} \tag{15}$$

They constitute a partition of D such that

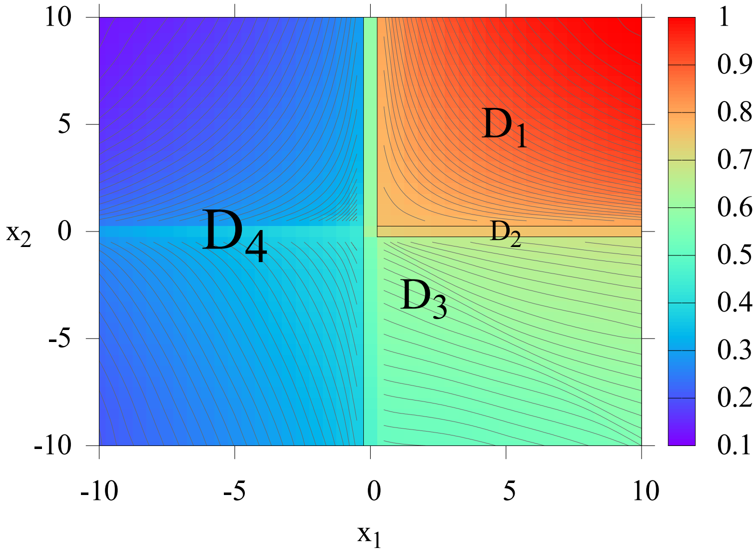
$$v(x) > v(y) > v(z) > v(w) \text{ for any } (x, y, z, w) \in D_1 \times D_2 \times D_3 \times D_4. \tag{16}$$

i.e., the elements of $\cup_{k=1}^4 D_k$ are coarse equivalence classes for v. The straight black lines in Figure 1 represent borders between those equivalence classes, and the gray curves represent indifference with respect to v. Plotting of indifference curves is suppressed along the zero bins of D_0 because v is discontinuous at such points. But the data show that v increases in x_1 for $x_2 = 0$, whereas for $x_1 = 0$, v increases as $x_2 \to 0$.

Any 2×2 game $G = (S, \pi)$ is solved with the representative solution concept F^2 by using the contour lines in Figure 1 to rank the strategy profiles $\mathbf{s} \in G$ by their deviation losses $\delta(\mathbf{s})$.

For many games, this ranking can be obtained directly from the partition $\{D_k\}_{k=1}^4$. Let us say that a game $G \in \Gamma^2$ *is solved in* E if $\delta(F^2(G)) \subset E$ for a subset E of D. Since F^2

Figure 1: Contour plot of the numerical representation v for the solution concept F^2 on the set $D = [-10, 10]^2$ of pairs of deviation losses. The subsets $\{D_k\}_{k=1}^4$ constitute a partition of D such that $v(x) > v(y) > v(z) > v(w)$ for any $(x, y, z, w) \in D_1 \times D_2 \times D_3 \times D_4$. Straight black lines represent boundaries between those partition elements, and gray curves represent indifference with respect to v.



solves games by maximizing $v(\delta(\cdot))$, the definition of $\{D_k\}_{k=1}^4$ and the relation (16) imply the following: First, games with strict Nash equilibria are solved in D_1. Second, any other game with one or more weak Nash equilibria is solved in D_2 if player 1 has a strict best reply in any such equilibrium. Otherwise the game is solved in D_3. Finally, since any game can be solved in $D \setminus D_4$ by choosing a best reply for player 1 to any conjecture about player 2, it follows that no game is solved in D_4 and that all games without pure Nash equilibria are solved in D_3. This implies that the solution concept F^2 is rational in the sense that its action is always a best reply to its conjecture.

When determining the ASC for each run separately, one finds some degree of heterogeneity. This heterogeneity is not captured by the representative solution concept F^2 which is calculated using all solution concepts from all runs. To assess this variation, we consider the standard deviation of the rank scores $v(x)$ for each point x in the grid D_0 across runs.[13] On

[13]This measure excludes any variation among the agents in each run. However, as shown in Table 15, there is very little intra-run heterogeneity among the agents.

average, across all points in the grid, the standard deviation amounts to 0.07. However, on $D_0 \cap D_1$ it is only 0.0004. Hence the ordering of strategy profiles on $D_0 \cap D_1$ is essentially the same across runs. A clue to the nature of this common ordering is provided by the example good reply function $f^a(x)$ in (12), which evaluates to $x_1 \cdot x_2$ for $x > 0$. Indeed, $v(x_1, x_2)$ has a rank correlation of 0.9999 with the product $x_1 \cdot x_2$ across the set of points $x \in D_0 \cap D_1$.

These results show that the representative solution concept F^2 and (in light of the above robustness findings) the ASC, agree with the Harsanyi-Selten risk dominance concept (8) for 2×2 games with one or more strict Nash equilibria.

3.3.2 General games

We next extend F^2, the representative solution concept for 2×2 games, to the set of all bimatrix games by showing that the ASC has a good reply function with the following structure:

$$g(\mathbf{x}) = \sum_{k=1}^{K} (\alpha + \beta x_k), \tag{17}$$

where $\beta \neq 0$. Negative β's may occur because the signed effect of the arguments to g is determined by the composite function $f(g(\cdot), g(\cdot))$.

A good reply function g is said to be *additive* if (17) holds. To test if the g-function of the ASC is additive, we proceed as follows: For each agent a, generate a data set with 100 observations (y^a, x_1, x_2), where (x_1, x_2) is a vector of two random deviation losses and $y^a = g^a(x_1, x_2)$. Then estimate the linear model

$$y^a = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2, \tag{18}$$

and conclude that agent a has an additive good reply function if $\beta_1 = \beta_2$, $\beta_{12} = 0$, and the R^2 of the regression exceeds 0.99. For each run, compute the mean R^2 and the median values of the parameter estimates across all agents.[14] This yields a data set of 100 observations

[14]We use medians to aggregate the parameter estimates, because they can potentially vary widely across agents. But most agents have R^2's close to 1, so we use means to obtain conservative averages of R^2.

which is described in Table 17.

Table 17: Test of additive good reply functions by means of (18). The number of observations is 100, one observation for each run with the model. P-values (from left to right) refer to Wilcoxon tests against the null hypotheses that $\alpha = 0$; $\beta_1 = \beta_2$; and $\beta_{12} = 0$.

Parameter	α	β_1	β_2	β_{12}	R^2
Min	-0.047	1	1	0	0.999
Max	0.015	15	15	$8.2e{-}05$	1
Median	0	5	5	0	1
Mean	0.000	5.760	5.760	$8.4e{-}07$	1.000
P-value	0.394	0.995		0.371	

The table shows that β_1 is not significantly different from β_2; that β_{12} does not differ significantly from zero; and that $R^2 > 0.99$ for all runs. We can therefore conclude that additive good replies is a characteristic feature of the aggregate solution concept. Furthermore, since α does not differ significantly from 0, we can use the function g, which takes a variable number of arguments, to process those arguments by first summing them up and then applying the one-dimensional component of g to that sum. This means that Figure 1 can be used to solve all games by ranking the pairs of sums of deviation losses that correspond to each strategy profile of the game.

3.4 Robustness checks

The ASC is tested for robustness with respect to two changes to the model specification. First we consider the algorithm which computes good replies and ask if initialization by zero values could have introduced a bias towards additive good replies. Second we analyze to what extent individual solution concepts are affected by the presence of strictly dominated strategies. This yields the four experiments shown in Table 18. The case D0 is the one considered so far.

Table 18: Robustness checks

Strictly dominated strategies	Initial memory	
	Zero	Random
Allowed	D0	DR
Not allowed	N0	NR

3.4.1 Memory initialization

The algorithm in Table 2, computing good reply scores for strategy profiles, initializes its memory slots $\mathbf{z}$ to zero. On exit from the algorithm, the first memory slot z_1 contains its return value, which is taken to be the good reply score for the given strategy profile. In this setting, additive good reply function can be obtained as a single instruction which simply adds the next deviation loss to z_1. To gauge the extent to which the existence of this 'shortcut' may have influenced the results, we re-run the model with the memory slots initialized to random values. This experiment, called DR in Table 18, turns out to sometimes produce a new type of agent with *multiplicative* good reply functions of the following form:

$$g(\mathbf{x}) = \begin{cases} \prod_{k=1}^{K}(\alpha + \beta x_k), & \text{if } \mathbf{x} \geq 0 \\ \xi(\mathbf{x}) < \min\limits_{\mathbf{x}' \geq 0} \prod_{k=1}^{K}(\alpha + \beta x'_k), & \text{otherwise.} \end{cases} \tag{19}$$

The first case assigns high scores to vectors of deviation losses $\mathbf{x}$ that correspond to best replies, and the second case assigns low scores ($\xi(\mathbf{x})$) to all other $\mathbf{x}$. This suggests that solution concepts with multiplicative good replies are geared towards solving games at pure Nash equilibria whenever they exist.

The functions γ^a and f^a for a typical agent a of this type are listed in (20)–(21).

$$\gamma^a(x_k, z, k) \;=\; \begin{cases} 3.7 \cdot 10^{13} \cdot (1.7 \cdot 10^{-4} + x_k) \cdot \max(0, 1), & \text{if } k = 1 \\ 3.7 \cdot 10^{13} \cdot (1.7 \cdot 10^{-4} + x_k) \cdot \max(0, z), & \text{if } k > 1. \end{cases} \tag{20}$$

$$f^a(x_1, x_2) \;=\; \begin{cases} x_1 \cdot x_2, & \text{if } (x_1, x_2) > 0 \\ x_1, & \text{if } x_1 > 0 \text{ and } x_2 \leq 0 \\ 0, & \text{if } x_1 \leq 0. \end{cases} \tag{21}$$

At the first iteration ($k = 1$) of γ^a, it deals with the initial random z by replacing it with 1. After K iterations, one obtains the good reply function g^a in (22), which has the structure (19).

$$g^a(\mathbf{x}) = \begin{cases} \prod_{k=1}^{K} (6.3 \cdot 10^9 + 3.7 \cdot 10^{13} x_k) \geq (6.3 \cdot 10^9)^K, & \text{if } \mathbf{x} \geq 0 \\ 0, & \text{if } x_k < 0 \text{ for some } k < K \\ \prod_{k=1}^{K} (6.3 \cdot 10^9 + 3.7 \cdot 10^{13} x_k) < 0, & \text{if } \mathbf{x}_{-K} \geq 0 \text{ and } x_K < 0. \end{cases} \tag{22}$$

Pure Nash equilibria are represented by Case 1 of (21). Again we see that the good solution function scores such strategy profiles by the product of the good reply scores. Parallel to (11) of Section 3.3, the small constant in (20) guarantees that the function $f^a(g^a(\cdot), g^a(\cdot))$ extends continuously from strict to weak best replies because $(1.7 \cdot 10^{-4} + x_k)$ is positive for $x_k \geq 0$ and negative for all other deviation payoffs. This is shown below along with a proof of (22).[15]

The large constant factor in (20) is typical for multiplicative good reply functions. Larger

[15] We show that (22) holds. The first case follows directly from (20). Consider next the second and third cases of (22). Recall from footnote 12 that $\text{abs}(x_k) \leq 10^{-2}$ if $x_k \neq 0$. Consequently, the term $(1.7 \cdot 10^{-4} + x_k)$ in (20) is positive if $x_k \geq 0$ and negative otherwise, and $\gamma^a(x_k, \ldots) \leq 0$ if $x_k < 0$. If $x_k < 0$ for $k < K$, the term $\max(0, z)$ in (20) ensures that $\gamma^a(x_{k'}, \ldots) = 0$ for all $k' > k$, hence $g^a(\mathbf{x}) = 0$, which proves case 2 of (22). But if $x_k \geq 0$ for $k < K$ and $x_K < 0$ then $\gamma^a(x_k, \ldots) > 0$ for all $k < K$ and $g^a(\mathbf{x}) = \gamma^a(x_K, \ldots) < 0$, which proves case 3 of (22). Thus the random order in which deviation losses are presented to γ^a can lead to a negative or a zero score if one or more deviation losses are strictly negative. But the agent still behaves in a consistent manner because the good solution function f^a in (21) does not distinguish between zero and negative arguments.

constants cause more games to be solved at pure Nash equilibria, by producing larger g–scores which increase the likelihood that f^a reaches a maximum at case 1 of (21).[16] As an upshot, games without pure Nash equilibria are solved somewhat arbitrarily: The g–score to player 1 is maximized without regard for that of player 2 (case 2 of (21)). By comparison, the additive solution concept in (12)–(13) of Section 3.3 solves games without pure Nash equilibria by maximizing the g–score to player 2 on the set of strategy profiles that yield a positive g–score for player 1.

We do 100 runs with experiment DR and use the same procedure as in Section 3.3.2 to test for additive and multiplicative good replies: For each agent a, we generate a data set with 100 observations (y^a, x_1, x_2), where (x_1, x_2) is a vector of two random deviation losses and $y^a = g^a(x_1, x_2)$. Unlike in Section 3.3.2, we restrict the deviation losses to be positive and bounded away from 0 because we cannot exclude the possibility that some good reply functions have discontinuities close to zero values of the arguments, cf. (19) and (22). We then estimate the linear model (18) for each agent and aggregate the parameter estimates by runs.

A summary of the results is contained in Table 19, where $P(\beta_{12})$ is the P-value associated with the multiplicative term β_{12} in (18). We sort the sample by $P(\beta_{12})$, split it at $P(\beta_{12}) = 0.1$, and examine the estimated parameters to find that the 48 runs in Panel A (with parameters $\alpha = 0$, $\beta_1 = \beta_2$, $\beta_{12} = 0$ and $R^2 > 0.99$) have additive good reply functions, while the 52 runs in Panel B (with $\beta_{12} \neq 0$) are consistent with the multiplicative good reply structure in (19).

Table 19 confirms our conjecture that multiplicative good replies are closely associated with Nash equilibrium play. Variable *playNash* is the frequency of Nash equilibrium play in games with one Nash equilibrium, as explained in Section 3.2. On average, agents with multiplicative good replies play the Nash equilibrium 99% of the time, against 83.7% for the agents with additive good replies. By looking at individual runs, we find that 46 of the 52

[16] To see this, let (x_1, x_2) and (x_1', x_2') satisfy the conditions of case 1 and 2 of (21), respectively. Then $\lambda x_1 \cdot \lambda x_2 > \lambda x_1'$, for sufficiently large λ.

Table 19: Test of good reply functions by means of (18) for experiment DR. The number of observations is 100, one observation for each run with the model. Parameters α, β_1, β_2, β_{12} and R^2 are defined as in Table 17. $P(\beta_{12})$ is the median P-value by run associated with β_{12}, and *playNash* is the frequency of Nash equilibrium play in games with one Nash equilibrium. P-values (from left to right) refer to Wilcoxon tests against the null hypotheses that $\alpha = 0$; $\beta_1 = \beta_2$, and $\beta_{12} = 0$.

Panel A: $P(\beta_{12}) > 0.1$, 48 runs, none with *playNash* ≥ 0.99.							
Parameter	α	β_1	β_2	β_{12}	R^2	$P(\beta_{12})$	*playNash*
Min	-0.013	-6	-6	$-6.4e{-}09$	0.997	0.606	0.828
Max	1.000	18	18	$3.4e{-}08$	1.000	1.000	0.846
Median	0	2	2	0	1.000	1.000	0.838
Mean	0.043	3.667	3.667	$8.8e{-}10$	0.999	0.927	0.837
P-value	0.363	1.000		0.236			

Panel B: $P(\beta_{12}) \leq 0.1$, 52 runs, 46 with *playNash* ≥ 0.99.							
Parameter	α	β_1	β_2	β_{12}	R^2	$P(\beta_{12})$	*playNash*
Min	$-2.1e{+}27$	$-1.7e{+}26$	$1.1e{-}01$	$-7.9e{-}03$	0.161	0.000	0.838
Max	$4.1e{+}31$	$5.3e{+}30$	$3.7e{+}30$	$4.5e{+}29$	0.929	0.051	1.000
Median	$2.8e{+}17$	$8.3e{+}19$	$4.9e{+}21$	$2.7e{+}20$	0.659	0.000	0.997
Mean	$1.0e{+}30$	$1.6e{+}29$	$1.5e{+}29$	$1.5e{+}28$	0.593	0.001	0.990
P-value	0.000	0.449		0.000			

runs in Panel B have *playNash* ≥ 0.99 and that all 48 runs in Panel A have *playNash* < 0.85. The remaining 6 runs seem to represent a mix of agents with additive and multiplicative and good reply functions.

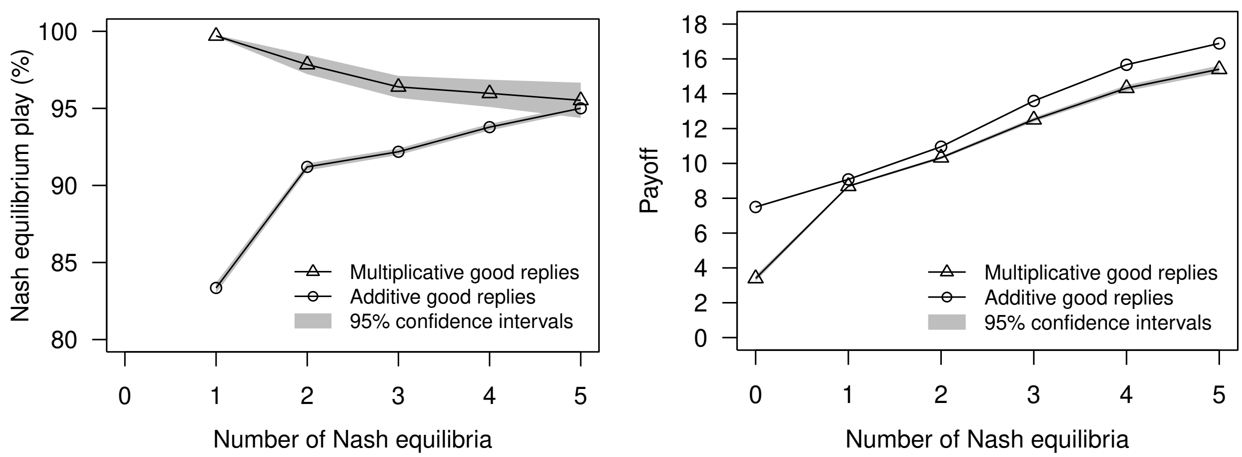
In what follows, we will disregard those 6 runs and reserve the term *multiplicative* for agents and model runs with *playNash* ≥ 0.99. Analogously, an *additive agent* is one whose good reply function yields $\beta_1 = \beta_2$, $\beta_{12} = 0$ and $R^2 > 0.99$ when fitted to (18), and an additive model run is one for which the median β's and the mean R^2 satisfy these conditions.

In the remainder of this subsection, we compare the 48 additive runs of Table 19, Panel A with the 46 multiplicative runs from Panel B. The 48 additive runs constitute an additive ASC, and the 46 multiplicative runs form a multiplicative ASC.

Applying the multiplicative ASC to the games in Section 3.1, we find less cooperation and lower aggregate payoffs as compared to the additive one: The 'refinement' games in Tables 8 and 9 are both solved at the inferior equilibrium (T, R), and in the Centipede game the agents take the money at the first opportunity. In ultimatum games with 5 or 10 dollars to share, the players offer and demand one dollar, and with 50 or 100 dollars to share, offers and demands amount to only 8% of the total.

We next compare the additive and the multiplicative ASC across a large number of games with a varying number of pure Nash equilibria. We create six sets of 1,000 games with the number of pure Nash equilibria ranging from 0 to 5 and solve each one of those 6,000 games with the two ASC's. The results are presented in Figure 2.

Figure 2: Behavior of the additive and the multiplicative aggregate solution concepts from experiment DR in games with a varying number of pure Nash equilibria. The number of observations is 94.



The left panel of Figure 2 shows the frequency of Nash equilibrium play.[17] In games with one pure Nash equilibrium, the multiplicative agents play that strategy profile in 99.7% of those games. As the number of pure Nash equilibria increases, the frequency of Nash play declines, but remains above 95%. The additive agents are not equipped to identify Nash equilibria. Instead they look for strategy profiles with positive sums of deviation losses, which become more prevalent as the number of pure Nash equilibria increases. In games with one pure Nash equilibrium, these agents play Nash only 84% of the time, but this frequency is increasing in the number of equilibria. For games with 5 pure Nash equilibria there is no significant difference between the two ASC's with respect to the frequency of

[17]If agents would independently randomize between the n row and n column strategies that support n pure Nash equilibria, the generic probability of playing some Nash equilibrium is $1/n$.

Nash equilibrium play.

The right panel of Figure 2 plots payoffs against the number of pure Nash equilibria for the two ASC's. Payoffs increase as the number of Nash equilibria increases, with additive agents doing better throughout. The difference is small for games with one pure Nash equilibrium, but widens as the number of equilibria increases. The multiplicative agents fare particularly badly in games with no pure Nash equilibrium, obtaining less than half the payoff of the additive agents.

3.4.2 Rationalizability

We have seen in Section 3.1 that the additive solution concept sometimes produces solutions that are not subgame perfect, or not Nash, or include strictly dominated strategies. While strictly dominated solutions agree with intuition or experiments for some games, it raises the issue to what extent the solution concept is robust with respect to addition of dominated strategies.[18] To illustrate the issue, we consider the game in Table 20.

Table 20: A game with strictly dominated strategies. Numbers in italics are probabilities (%).

(s,t)		A	b	c	
	σ	*0*	*100*	*0*	Col
A	*0*	**1**, **1**	11, 0	-1, -2	*0*
b	*100*	0, 11	10, 10	-2, 0	*100*
c	*0*	-2, -1	0, -2	-3, -3	*0*
	Row	*0*	*100*	*0*	ϕ

The game is symmetric and has one Nash equilibrium in pure strategies at (A,A), with payoffs $(1,1)$. The additive ASC solves the game at (b,b), which yields payoffs $(10,10)$. Human players might also be able to solve the game at (b,b) because it yields high, identical payoffs and only weak incentives to deviate to A. But this is not quite how the ASC arrives at its solution: When the good solution function takes sums of deviation losses as inputs, (b,b)

[18] Kohlberg & Mertens (1986) dismiss the idea of robustness with respect to addition of strictly dominated strategies in relation to strategic stability, but in our case, there are additional considerations to be made.

is selected because it has a high g-score of 9 because $(10-11)+(10-0) = -1+10 = 9$. The smaller negative term -1 is associated with the weak incentives to deviate. But the larger positive term 10 is due to the presence of the dominated action c. Although the ASC seems to have found the 'right' solution to this game, it may have done so for the wrong reason. If the dominated action c is eliminated from the game, we obtain a Prisoner's dilemma game which is solved at (A, A) by the ASC, cf. Section 3.1.

It is easy to construct this type of examples by adding strictly dominated strategies to an existing game. An obvious remedy would be to iteratively eliminate strictly dominated strategies (IESDS) before presenting the game to the ASC for solution. The modified ASC would then solve the game in Table 20 at (A, A) and any other game at some rationalizable pair of strategies. However, the ASC may no longer be stochastically stable if IESDS is imposed on it ex post. We will therefore impose IESDS ex ante and see if, and how, this affects the aggregate solution concept.

To that end, we carry two additional experiments, N0 and NR, each one consisting of 100 runs with the model. N0 and NR are identical to D0 and DR, respectively, except that the agents are not allowed to play strictly dominated strategies, see Table 18. This restriction is imposed by iteratively removing all strictly dominated strategies from any game before applying some solution concept.

Experiment NR (IESDS and random initial memory) yields 81 additive runs and one multiplicative one out of 100 runs in total. Further, 100 runs of experiment N0 (IESDS and zero initial memory) yields 93 additive runs and no multiplicative ones. Thus IESDS strengthens the additive solution concept by removing some potentially irrelevant information which the additive solution concept is unable to detect. Apparently, this effect is strong enough, or the competition from multiplicative Nash players is weak enough, for the additive solution concept to prevail when IESDS is imposed.

4 Conclusion

The paper uses a genetic programming algorithm to study evolution of play in one-shot bimatrix games. The model has 2,000 artificial agents who gain experience with one-shot play across 100,000 periods. In each period, each agent plays 1,000 random bimatrix games with 2-10 strategies per player in random positions (row or column) against random opponents. To play games, each agent uses an individual solution concept, which can be thought of as a soft, non-equilibrium generalization of the Nash equilibrium concept. The individual solution concepts admit a numerical representation in terms of two functions: The first one assigns a good reply score to each strategy profile based on a player's deviation losses, and the second one aggregates both players' good reply scores to obtain a measure of the degree to which a strategy profile constitutes a good solution. By taking the mean of all individual solution concepts for each game we obtain an aggregate solution concept (ASC).

We do 100 runs with the model and show that the ASC converges to a stochastically stable equilibrium. The individual solution concepts turn out to have a common structure with simple additive good reply functions and complex good solution functions. The good solution functions produce coarse orderings of their domains based on the signs of the good reply scores, and a continuous numerical ranking on each equivalence class of that ordering. In particular, for positive pairs of good reply scores, the good solution score is the product of those pairs. This yields risk dominance for 2×2 games and an extension of that solution concept to games with higher dimensions.

Applying this ASC to a number of well-known games, we find that it agrees well with intuition and empirical evidence. Examples include the Ultimatum game, the Traveler's dilemma, the Centipede game and a collection of games from the refinement literature. It also behaves as if the agents were motivated by social norms in some games that were designed to test such concepts as fairness, trust and reciprocity. In our model, such results are due to positive correlation between payoffs and deviation losses, and a solution concept which resembles a social welfare function by solving many games at strategy profiles which maximize the product of the players' sums of deviation losses.

We test the robustness of the main result by varying some aspects of the model specification. One such model variant produces an approximate 50–50 distribution of two different solution concepts. One half has the additive good reply functions of the base case, and the other half has a new type of multiplicative good reply functions. The latter play Nash equilibria more often than the former. In games with one pure Nash equilibrium the frequency of Nash play is almost 100% for the multiplicative solution concept, as compared to 84% for the additive one. However, on games without pure Nash equilibria, the multiplicative solution concept does not perform well, and in all other model variants, the multiplicative solution concept is virtually absent.

Our approach to modeling one-shot play can be extended in several directions. (1) We have imposed fairly tight restrictions on the solution concepts in order to stay close to Nash, and some of those restrictions can be relaxed. For example, we assumed that payoffs are von Neumann–Morgenstern utilities and imposed Invariance with respect to positive affine payoff transformation to reflect that assumption. Dropping it would be a first step towards building a model with monetary payoffs, and one way to proceed from there would be to evolve utility functions along with the good reply and good solution functions. (2) Our agents are boundedly rational due to computational constraints on program length (32 instructions) and scratch memory (4 memory slots). These parameters can be varied to study behavioral effects of variations in bounded rationality. (3) By representing games in terms of of vectors of deviation losses, our model forces the agents to focus on strategic stability, i.e., variations in player i's payoffs for a given action by player j, with no focus on risk, i.e., variations in i's payoffs for a given action by player i. In experiments with human subjects, such risk considerations seem to play a role, and it would be of interest to see if our artificial agents would make the same considerations if they were provided with the relevant information.

References

Arifovic, J. (1994). Genetic algorithm learning and the cobweb model, *Journal of Economic Dynamics and Control* **18**(1): 3–28.

Arifovic, J. (1995). Genetic algorithms and inflationary economies, *Journal of Monetary Economics* **36**(1): 219–243.

Arifovic, J. (1996). The behavior of the exchange rate in the genetic algorithm and experimental economies, *Journal of Political Economy* **104**(3): 510–541.

Basu, K. (1994). The Traveler's dilemma: Paradoxes of rationality in game theory, *American Economic Review: Papers and Proceedings* **84**(2): 391–395.

Bernheim, B. D. (1984). Rationalizable strategic behavior, *Econometrica* **52**(4): 1007–1028.

Brown, G. W. (1951). Iterative solution of games by fictitious play, *Activity Analysis of Production and Allocation* **13**(1): 374–376.

Capra, C. M., Goeree, J. K., Gomez, R. & Holt, C. A. (1999). Anomalous behavior in a traveler's dilemma?, *American Economic Review* **89**(3): 678–690.

Carlson, H. & van Damme, E. (1993). Equilibrium selection in stag hunt games, *in* K. Binmore, A. Kirman & P. Tani (eds), *Frontiers of Game Theory*, Addison-Wesley, Reading, MA, pp. 237–254.

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*, Springer, pp. 571–598.

Cooper, D. J. & Kagel, J. H. (2003). Lessons learned: Generalizing learning across games, *American Economic Review* **93**(2): 202–207.

Cooper, D. J. & Kagel, J. H. (2008). Learning and transfer in signaling games, *Economic Theory* **34**(3): 415–439.

Costa-Gomes, M., Crawford, V. P. & Broseta, B. (2001). Cognition and behavior in normal-form games: An experimental study, *Econometrica* **69**(5): 1193–1235.

Crawford, V. P., Costa-Gomes, M. A. & Iriberri, N. (2013). Structural models of nonequilibrium strategic thinking: Theory, evidence, and applications, *Journal of Economic Literature* **51**(1): 5–62.

Fudenberg, D. & Liang, A. (2019). Predicting and Understanding Initial Play, *American Economic Review* **109**(12): 4112–4141.

Gale, J., Binmore, K. G. & Samuelson, L. (1995). Learning to be imperfect: The ultimatum game, *Games and Economic Behavior* **8**(1): 56–90.

Germano, F. (2007). Stochastic evolution of rules for playing finite normal form games, *Theory and Decision* **62**(4): 311–333.

Gilboa, I. & Schmeidler, D. (1995). Case-based decision theory, *Quarterly Journal of Economics* **110**: 605–639.

Gilboa, I., Schmeidler, D. & Wakker, P. P. (2002). Utility in case-based decision theory, *Journal of Economic Theory* **105**(2): 483–502.

Grimm, V. & Mengel, F. (2012). An experiment on learning in a multiple games environment, *Journal of Economic Theory* **147**(6): 2220–2259.

Güth, W. R., Schmittberger, R. & Schwarze, B. (1982). An experimental analysis of ultimatum bargaining, *Journal of Economic Behaviour & Organization* **3**(4): 367–388.

Güth, W. R. & Tietz, R. (1990). Ultimatum bargaining behavior: A survey and comparison of experimental results, *Journal of Economic Psychology* **11**(3): 417–449.

Harsanyi, J. & Selten, R. (1988). *A General Theory of Equilibrium Selection in Games*, MIT Press.

Haruvy, E. & Stahl, D. O. (2012). Between-game rule learning in dissimilar symmetric normal-form games, *Games and Economic Behavior* **74**(1): 208–221.

Jehiel, P. (2005). Analogy-based expectation equilibrium, *Journal of Economic Theory* **123**(2): 81–104.

Kohlberg, E. & Mertens, J.-F. (1986). On the strategic stability of equilibria, *Econometrica* **54**(5): 1003–1037.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

Koza, J. R. (ed.) (1992-2003). *Genetic Programming Series*, Kluwer Academic Publishers.

Lensberg, T., Schenk-Hoppé, K. R. & Ladley, D. (2015). Costs and benefits of financial regulation: Short-selling bans and transaction taxes, *Journal of Banking & Finance* **51**: 103–118.

LiCalzi, M. (1995). Fictitious play by cases, *Games and Economic Behavior* **11**(1): 64–89.

Marks, R. E. (2002). Playing games with genetic algorithms, *in* S. H. Chen (ed.), *Evolutionary Computation in Economics and Finance. Studies in Fuzziness and Soft Computing*, Vol. 100, Physica, Heidelberg, pp. 31–44.

McKelvey, R. D. & Palfrey, T. R. (1992). An experimental study of the centipede game, *Econometrica* **60**(4): 803–836.

Mengel, F. (2012). Learning across games, *Games and Economic Behavior* **74**(2): 601–619.

Mookherjee, D. & Sopher, B. (1994). Learning behavior in an experimental matching pennies game, *Games and Economic Behavior* **7**(1): 62–91.

Nagel, R. (1995). Unraveling in guessing games: An experimental study, *American Economic Review* **85**(5): 1313–1326.

Nordin, P. (1997). *Evolutionary Program Induction of Binary Machine Code and its Applications*, Krehl Verlag, Münster.

Pace, M. (2009). How a genetic algorithm learns to play traveler's dilemma by choosing dominated strategies to achieve greater payoffs, *2009 IEEE Symposium on Computational Intelligence and Games*, IEEE, pp. 194–200.

Pearce, D. G. (1984). Rationalizable strategic behavior and the problem of perfection, *Econometrica* **52**(4): 1029–1050.

Rosenthal, R. W. (1981). Games of perfect information, predatory pricing and the chain-store paradox, *Journal of Economic Theory* **25**(1): 92–100.

Roth, A. E. & Erev, I. (1995). Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term, *Games and Economic Behavior* **8**(1): 164–212.

Samuelson, L. (2001). Analogies, adaptation, and anomalies, *Journal of Economic Theory* **97**(2): 320–366.

Selten, R. (1967). Die Strategiemethode zur Erforschung des eingeschränkt rationalen Verhaltens im Rahmen eines Oligopolexperiments, *in* H. Sauermann (ed.), *Beiträge zur experimentellen Wirtschaftsforschung*, Mohr, Tübingen, pp. 136–168.

Selten, R., Abbink, K., Buchta, J. & Sadrieh, A. (2003). How to play (3×3)-games. A strategy method experiment, *Games and Economic Behavior* **45**(1): 19–37.

Sgroi, D. & Zizzo, D. J. (2009). Learning to play 3×3 games: Neural networks as bounded-rational players, *Journal of Economic Behavior & Organization* **69**(1): 27–38.

Spiliopoulos, L. (2011). Neural networks as a unifying learning model for random normal form games, *Adaptive Behavior* **19**(6): 383–408.

Spiliopoulos, L. (2015). Transfer of conflict and cooperation from experienced games to new games: A connectionist model of learning, *Frontiers in Neuroscience* **9**(102): 1–18.

Stahl, D. O. (1996). Boundedly rational rule learning in a guessing game, *Games and Economic Behavior* **16**(2): 303–330.

Stahl, D. O. (1999). Evidence based rules and learning in symmetric normal-form games, *International Journal of Game Theory* **28**(1): 111–130.

Stahl, D. O. (2000). Rule learning in symmetric normal-form games: Theory and evidence, *Games and Economic Behavior* **32**(1): 105–138.

Stahl, D. O. (2001). Population rule learning in symmetric normal-form games: Theory and evidence, *Journal of Economic Behavior & Organization* **45**(1): 19–35.

Stahl, D. O. & Wilson, P. W. (1994). Experimental evidence on players' models of other players, *Journal of Economic Behavior & Organization* **25**(3): 309–327.

Steiner, J. & Stewart, C. (2008). Contagion through learning, *Theoretical Economics* **3**(4): 431–458.

Taylor, P. D. & Jonker, L. B. (1978). Evolutionary stable strategies and game dynamics, *Mathematical biosciences* **40**(1-2): 145–156.

Van der Heijden, E. C., Nelissen, J. H., Potters, J. J. & Verbon, H. A. (1998). The poverty game and the pension game: The role of reciprocity, *Journal of Economic Psychology* **19**(1): 5–41.

Young, H. P. (1994). The evolution of conventions, *Econometrica* **61**(1): 57–84.