Evolutionary stable solution concepts for initial play

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

August 6, 2019

Abstract

We model initial play in bimatrix games by a large population of agents. The agents have individual solution concepts (maps from games to strategy profiles) that they use to solve games. In contrast to evolutionary game theory, where the agents play the same game indefinitely, we consider a setting where they never play the same game twice. Individual solution concepts are represented as computer programs which develop over time by a process of natural selection. We derive an aggregate solution concept (ASC), which converges to a stochastically stable state where the population mean behavior remains constant. The logic and performance of the evolutionary stable ASC is examined in detail, and its solutions to many well-known games are held up against the theoretical and empirical evidence. For example, the ASC selects the "right" solution to traveler's dilemma games, and predicts that the responder will get 40% of the pie in ultimatum games.

Keywords: Initial play, evolutionary stability, solution concepts, strategy method.

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

Initial play refers to situations where the players have no previous experience with the game they are about to play. Such situations pose interesting challenges to game theory. Theoretically, the existence of multiple Nash equilibria raises the question of which one, if any, of those equilibria will be played. Empirically, there is ample evidence that equilibrium solution concepts fail to predict actual behavior in many games.

Binmore (1987) introduced the terms *eductive* and *evolutive* to distinguish between the two main approaches to game theory. Eductive game theory focuses on the mental processes of rational agents who attempt to "forecast the forecasts of others". Eductive models of initial play include level-k reasoning, which can account for non-equilibrium behavior in many games, and the large literature on equilibrium refinements, which aims to sharpen predictions in games with multiple Nash equilibria. The evolutive approach includes evolutionary game theory and other kinds of learning processes in repetitive environments. Because initial play deals with one-shot games, there have been few attempts to apply the evolutive approach to the problem of initial play.

One notable exception is Selten, Abbink, Buchta & Sadrieh (2003). The paper provides a detailed account of an experiment aimed at studying initial play in 3x3 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 3x3 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. During the experiment, the students' programs played pure Nash equilibria with increasing frequency, and with a preference for maximal joint payoffs in games with multiple equilibria. In the absence of pure Nash equilibria, the programs mostly followed a best-reply cascade, as in level-k reasoning.

Our inquiry is similar to that of Selten et al. (2003). We will use the strategy method to run experiments with many agents who use individual solution concepts, represented as computer programs, to solve games. But our inquiry is more ambitious as we will consider the class of all bimatrix games and solution concepts that apply to all games in this class. We are interested in distributions of the individual solution concepts in large populations of agents, in particular whether the distributions have certain equilibrium and robustness properties.

By taking the mean across all individual solution concepts for each game, we obtain an aggregate solution concept (ASC). As a result of our inquiry we hope to find an ASC that forms a stochastically stable equilibrium (SSE). An SSE is 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 is something that could survive as a real world phenomenon.

To find an SSE we could proceed, as Selten et al. (2003), with a large number of untiring human players who gather experience with initial play and use it to improve their computer programs over time. We would then trace the evolution of the aggregate solution concept; occasionally add some noise and wait for an SSE to emerge. As such an approach seems impractical given the complexity of the task, we shall automate the process by employing a technique from computer science.

Genetic programming is a technique for "programming computers by means of natural selection" (Koza 1992). In our context it 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 the programs play another random set of games. By continuing in this manner across thousands of iterations, the programs become increasingly better at initial play until possibly, the process converges to a stochastically stable equilibrium.

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.

In Selten et al.'s (2003) experiment, the students had to make sure that their programs would be able to recommend an action for each position (row and column) in every possible 3x3 game. Each such program therefore qualifies as an (individual) *solution concept* for that class of games. In Nash and other equilibrium solution concepts, a pair of actions solves a game if it does so from both players' point of view. In the experiment, this equilibrium condition was not imposed, possibly because "subjects in the laboratory [] seem to avoid circular concepts in their strategic reasoning." (Selten 1998). Instead, by presenting a game to a program from the row player's point of view, the program would come up with an action for the row player, and by presenting it from the column player's point of view, the program would recommend an action for the column player. For many games, the pairs of actions obtained in this way were not mutual best replies. However, by examining the computer programs the experimenters found that most of them played best replies against some *conjecture* about the action of the other player.

In this paper, we will also work with a family of individual solution concepts that allow for non-equilibrium behavior. To solve a game, it must then be solved from both players' point of view. The solution to any game from a particular player's point of view will consist of two elements: An action for that player and a conjecture about the action of the opponent. A solution concept is an equilibrium one if for all games, the actions of both players coincide with the conjectures of their opponents, otherwise it is a non-equilibrium solution concept. Solution concepts for initial play must allow for non-equilibrium behavior because the agents never get a chance to react to false conjectures.

We will take steps to enforce some degree of rationality on the solution concepts. Throughout, 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[1] solution concepts.

Some structure must be imposed on the agents' programs to make their task a manageable one. To that end, we provide each agent with two programs instead of one and apply a soft version of the logic used to find a Nash equilibrium in pure strategies. One program will compute the degree to which a strategy for a player is a *good reply* to a given strategy for the opponent, and the other will look at pairs of good reply scores and compute the degree to which a pair of strategies constitutes a *good solution*. To solve the game from the point of view of a particular player, one computes the good solution scores for all pairs of action and conjecture for that player and declares those pairs that yield the maximal score to be the set of solutions. To play the game, the agent randomly selects one solution and does the associated action. The associated conjecture entails no consequences for the agent, but can be used to understand the logic behind the agent's action.

The overall spirit of our approach is related to evolutionary game theory in the sense that we also apply the forces of natural selection. But it differs from evolutionary game theory with respect to the complexity of the objects that compete to be selected. In our setting, the competition takes place among mappings from a space of games to solutions in those games, whereas in the simpler setting of evolutionary game theory, it takes place among solutions to a given game.

Our approach is related to the literature on learning across games. In particular, it is

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

inspired by Gale, Binmore & Samuelson's (1995) evolutionary analysis of the Ultimatum game. Another early contribution is LiCalzi (1995), who studies sequences of random games of fixed dimension played by two players. By assuming that all games belong to one and the same equivalence class, learning across games can be represented by a modified fictitious play algorithm. For 2x2 games, the process is shown to converge under suitable restrictions on the algorithm. More recently, Germano (2007) constructs an evolutionary model of solution concepts for finite normal form games. The number of strategies is the same for each player in every game, and there is a finite number of solution concepts. This permits the construction of an average game which is then used to derive the main results. Mengel (2012) studies learning across games in finite sets of two-person games, all of which have the same number of strategies. The agents can partition the games into analogy classes, with finer partitions being more costly. An equilibrium (in partitions and actions) for the learning process provides an equilibrium selection mechanism for individual games which can select among some strict Nash equilibria and account for some deviations from subgame perfectness that have been observed in experiments with human subjects. An empirical test of Mengel's (2012) partition model is provided by Grimm & Mengel (2012) on data from experiments with human subjects.

Beginning with Stahl & Wilson (1994), there is a substantial experimental literature on initial play. In such experiments, one finds that the subjects often deviate in systematic ways from equilibrium play, and that structural non-equilibrium models such as level-k reasoning do a better job of predicting actual outcomes. Crawford, Costa-Gomes & Iriberri (2013) provide a survey of this literature. A recent contribution is Fudenberg & Liang (2019), who use machine learning to detect new regularities in the experimental data. In this paper, we use another form of machine learning for the different purpose of finding evolutionary stable solution concepts.

In experiments with initial play the subjects usually play a sequence of different games, but without intermediate feedback. The purpose is to preserve an impression of initial play by minimizing the effects of repetition. But a side effect of such treatments is that inexperienced subjects remain so during the whole experiment. This contrasts with Selten et al. (2003) and our paper, where the agents become experienced in initial play over time by receiving feedback after every round of play.

Our paper is exploratory, and there is no particular hypothesis to be tested. The aim is to see whether the approach works, and, if it does, what outcome it produces. Will we see a sensible solution concept emerging? Does it play Nash, all the time, most of the time, and when does it not? Will the solution concept select between multiple Nash equilibria in any reasonable way, and will it match human behavior in games where non-Nash play is prevalent?

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

2 Model

In this section we introduce a general class of solution concepts and a genetic programming (GP) algorithm to represent the evolution of increasingly better solution concepts. 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.

An equilibrium solution concept is one that solves any game G at (s,t) if and only if it solves its transpose at (t,s). That yields consistency of actions and conjectures, at least for games that have unique solutions. In our context of initial play, one cannot in general expect such consistency to come about because the agents never get a chance to react to false conjectures.

Non-equilibrium solution concepts. To model initial play, we need solution concepts that can account for non-equilibrium behavior. This means that games must be solved from both players' point of view. To formalize, 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 $(s,t) \in S$. A solution concept F is *non-equilibrium* if it admits the following interpretation:

1. Each $(s,t) \in F(G)$ is a solution to G from player 1's point of view. s is 1's action and t is her conjecture about player 2's action.

2. Each $(t', s') \in F(G^\top)$ is a solution to G from player 2's point of view. t' is 2's action and s' is his conjecture about player 1's action.

This definition generalizes the notion of an equilibrium solution concept by adding a player perspective and introducing an explicit distinction between actions and conjectures. It should be noted that a non-equilibrium solution concept admits non-equilibrium play without precluding equilibrium solution concepts. From now on, it will be assumed that all solution concepts are non-equilibrium ones. We next provide a formal description of how two agents will use such solution concepts to play games.

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 large 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 $P(s,t,G) = (P_1(s,t,G), P_2(s,t,G))$. For population A, the bimatrix $P(\cdot,\cdot,G)$ is the *aggregate solution* to game G, 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.

Mixed Nash equilibria. In this model, the agents solve games by choosing a pair of

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

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.

To illustrate, consider the game G in Table 2. Suppose agent a plays A as player 1 because she believes the opponent is a chicken who will certainly play C. Then the same reasoning[2] would lead her to play D as player 2, assuming that 1 will play B. Similarly, another agent b might do C as player 2 and B as player 1 because he suspects that someone like a is the opponent. Let F^a and F^b denote the solution concepts of agents a and b. By assumption, $F^a(G) = \{(A,C)\}$, $F^a(G^\top) = \{(D,B)\}$, $F^b(G) = \{(B,D)\}$ and $F^b(G^\top) = \{(C,A)\}$.

Table 2: A game G. Pure Nash equilibria in boldface.

(s,t)	C	D
A	$\mathbf{3},\mathbf{2}$	$0,0$
B	$0,0$	$\mathbf{2},\mathbf{3}$

Consider a population of 1,000 agents, composed of 600 copies of agent a and 400 copies of agent b. Game G is reproduced in Panel (a) of Table 3. Panel (b) contains the aggregate solution bimatrix $P = (P_1, P_2)$ along with its marginal distributions σ (mixed actions) and ϕ (conjectures) for the row and column players. For convenience, the mixed actions and conjectures are also shown along with the payoff matrix in Panel (a).

[2]Formally, the conclusion follows if solution concepts are invariant with respect to reorderings of strategies. We will impose this condition below under the name of "Symmetric good replies". For the game G in Table 2, by taking the transpose $G^\top$ and reordering the two rows and the two columns, one obtains a game with the same payoff matrix as G, and where the strategies for Row and Col are $\{D,C\}$ and $\{B,A\}$, respectively.

Table 3: Game G of Table 2 with aggregate solution, mixed actions (σ) and conjectures (ϕ). Numbers in italics are probabilities (%).

(a) Payoffs, actions and conjectures

(s,t)		C	D	
	σ	*40*	*60*	Col
A	*60*	**3**, **2**	0, 0	*40*
B	*40*	0, 0	**2**, **3**	*60*
	Row	*60*	*40*	ϕ

(b) Solution, actions and conjectures

(s,t)		C	D	
	σ	*40*	*60*	Col
A	*60*	*60*,*40*	*0*, *0*	*40*
B	*40*	*0*, *0*	*40*,*60*	*60*
	Row	*60*	*40*	ϕ

Panel (b) shows, for instance, that $P(A, C) = (60, 40)$. It means that (1) 60% of the row players do A, conjecturing that the column player will do C. These are the 600 agents who use solution concept F^a. (2) 40% of the column players do C, conjecturing that the row player will do A. These are the 400 agents who use solution concept F^b.

Suppose the agents are randomly matched to play the game. Then, according to the mixed actions σ in Table 3, A and D are played with 60% probability, and B and C are played with 40% probability. These actions form the mixed Nash equilibrium of game G. Strictly speaking, they form a correlated equilibrium (Aumann 1974) because we do not allow the agents to play games against themselves. For example, in a two-agent population with one a and one b the agents would always play (A, C) or (B, D), and so their actions would be perfectly correlated. However, in large populations, as will be considered here, the agents' actions will be (almost) uncorrelated.

Comparing the mixed actions and conjectures of Table 3, we see that they are not consistent. For instance, while 60% of the Col players conjecture that Row will do B, only 40% of the Row players actually do so. The inconsistency arises because the agents use non-equilibrium solution concepts. In principle, the agents are then free to conjecture anything they want, and it is this degree of freedom which enables the population of agents to play mixed Nash equilibria. In Table 3, C is the best reply for those 40% of Col players who conjecture that Row will do A, and D is the best reply for those 60% Col players who conjecture that Row will do B.

This example shows that it is possible for a population of agents to play mixed Nash equilibria without any external intervention, provided they are allowed to use non-equilibrium solution concepts. In Section 3, we shall see that the agents actually 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 will consider a class of representable solution concepts that includes the Nash equilibrium concept 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

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

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

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

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

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.

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 the 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). We next discuss whether some of them should 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 irrelevant aspects of the game.

Symmetric good solutions. A solution concept with symmetric good solutions is one that solves any game G at $(s,t) \in \Sigma(G)$ if and only if it solves its transpose $G^\top$ at (t,s). The Nash equilibrium concept satisfies this symmetry as well because f^N is symmetric. We will not impose this requirement because we will allow the agents to use non-equilibrium solution concepts.

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 4 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 4: 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 4. 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,[3] following Nordin (1997). Each program consists of at most 32 machine

[3]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

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 4) are preserved across iterations.

Program instructions operate on variables, numerical constants, and the memory slots. An instruction specifies 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.[4] 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.[5]

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 $i \in \{1, 2\}$.

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

humans and analyzed by computer algebra applications.

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

[5] 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.

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.[6]

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

Table 5: 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

[6] 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.

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 5, 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.[7]

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.

[7] 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.

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 initial play, 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 initial play, 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:

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 non-equilibrium 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 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 initial play 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 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. These two measures are also employed to check convergence. 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 of initial play by experienced agents.

3.1.1 Classical games

In 'Battle of the sexes', 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. In Section 2.1, we used this game to illustrate the role of the population in creating mixed strategies. Panel (a) of Table 6 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.

Panel (b) of Table 6 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).

Table 6: 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) 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.

The 'Rock, Paper, Scissors' game, depicted in Table 7 Panel (a), is a zero-sum game with 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 7: 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.

We next consider a game where the agents' self-interest prevails. In 'Prisoners' dilemma', Table 8, the players get a sentence depending on whether they deny (d) or confess (C) a crime. Deny is strictly dominated[8], (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.

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.

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

Table 8: 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

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.

In the market entry game, Table 9, 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 by bold type. Backward induction supports (E, A), and so does the ASC, which plays this pair of strategies with 97% probability.

Table 9: Market entry game. Numbers in italics are probabilities (%).

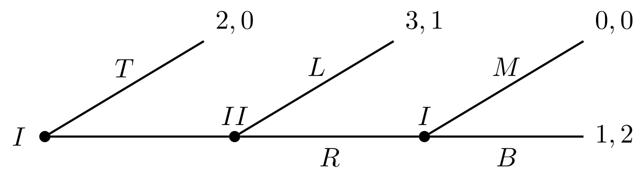
(s,t)		F	A	
	σ	*0*	*100*	Col
O	*3*	$\mathbf{2}, \mathbf{2}$	$2, 2$	*0*
E	*97*	$0, 0$	$\mathbf{3}, \mathbf{1}$	*100*
Row		*3*	*97*	ϕ

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

Kohlberg and Mertens I. The game in Table 10 has two pure Nash equilibria; (T, R) and (M, L). Backward induction selects (T, R), but the ASC selects (M, L), which is supported against (T, R) by a forward induction argument: At (T, R), B is an inferior response for Row, and by eliminating B, one obtains a game which is solved at (M, L) by backward induction. The same argument holds for all but one of the equilibria in the unique stable set

Table 10: Kohlberg & Mertens (1986, p. 1029). Numbers in italics are probabilities (%).

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

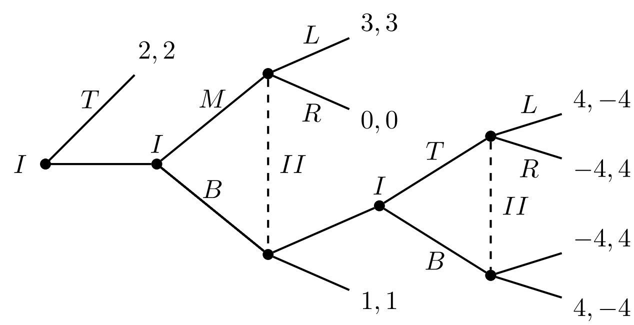


of this game, which equals the convex hull of (T, R) and $(T, \frac{1}{2}L + \frac{1}{2}R)$. But at $(T, \frac{1}{2}L + \frac{1}{2}R)$ the arguments fails, and therefore (T, R) is strategically stable while (M, L) is not.

Kohlberg and Mertens II. The game in Table 11 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. The payoff submatrix on $\{BT, BB\} \times \{LL, LR\}$ is a zero-sum subgame of the extensive form. By replacing the subgame by its value (0) and applying iterated dominance, one finds that the mixed equilibrium selected by the ASC is also the unique strategically stable set of this game.

Table 11: 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	$\mathbf{2},\mathbf{2}$	*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

Table 12: 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

a catch of smaller game amounting to a fraction $x \in (0, 1)$ of what each of them would get by cooperating.

The game is illustrated in Table 12. 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 12 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.

Table 13: Ultimatum game. Numbers in italics are probabilities (%).

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

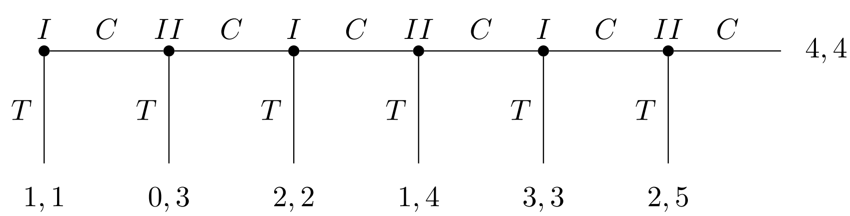
A small version of this game (with 5 dollars to share) is shown in Table 13. Action O_k for Row stands for 'Offer k dollars', and action A_k for Col stands for 'Accept any offer of k or more dollars'. In the ASC, Row offers 2 dollars, and Col accepts all offers of 2 or more. If the total amount is doubled 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 if the other decides to take, his payoff decreases. A version of this game is shown in Table 14. For each player, C_n denotes the strategy of n C's and then a T if $n < 3$.

Table 14: 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

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% 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 15, where C_n and c_n stand for 'Claim n dollars'.

When $R = R^* \equiv 2$, the agents make the minimal and maximal claims with equal probability, as shown in Table 15. When $R > R^*$, all agents claim the minimal 4, and when

Table 15: 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*	**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

$R < R^*$ 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 in each specific game. Thus again, as in the 'Rock, Paper, Scissors' game, we find that the model can lead to behavior as if the agents were driven by other motives than self-interest.

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

$$u_1(s,t) = (9-s)\cdot(1+t) \tag{9}$$

$$u_2(s,t) = (9-t)\cdot(1+s) \tag{10}$$

for players 1 and 2, respectively. The simultaneous move version of this game is shown in Table 16, 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 16: 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

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 17 contains descriptive statistics for a set of variables that measure the performance

Table 17: 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					
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%

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. *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. *playBestReply* and *meanPayoff* are computed separately for each game and each position and then averaged across all games and positions. *gainBR* is computed at an aggregate level because game payoffs are normally distributed with a zero mean.

The variables in Panel 2 of Table 17 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 17 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*, 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 three variables in Panel 1 of Table 17 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 18. The results are consistent with the hypothesis that the 100 model runs have reached stochastically stable equilibria after 80,000 iterations.

Table 18: Convergence tests. Tests of differences in means for the variables *playBestReply*, *meanPayoff* and *gainBR* across two intervals of model iterations. The number observations is 2,200 for each variable.

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

3.3 Agent program structure

In this section, we examine the computer programs of all of the agents across all runs with the aim of uncovering structural properties of the aggregate solution concept. Recall from Section 2.4 that each agent a is equipped with a pair of 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.

The ASC consists of 200,000 such pairs of programs. To find the structure of such a complex object we need some working hypotheses to start with. To that end, we pick agents randomly until we come across a pair of programs that boil down to something simple.[9] Such a pair of programs is listed in (11) and (12).

$$\gamma(x_k, z) = z + 15x_k + 0.023773 \tag{11}$$

$$f(x_1, x_2) = x_1 \cdot \max\left(\frac{0.5028277 + 0.3232822/x_1}{10^{16}x_2^6}, x_2\right). \tag{12}$$

Consider first the γ-function in (11). It starts with $z = 0$ and computes $z \leftarrow \gamma(x_k, z)$ for each element in a vector $\mathbf{x} = (x_1, ..., x_K)$ of deviation losses. When all the elements have

[9] We use the following procedure 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. If the resulting program is small (few non-NOPs), one can then translate the byte code into humanly readable language and use some computer algebra system to analyze it.

been processed, the g-score $g(\mathbf{x})$ is obtained as the terminal value of z, i.e.,

$$g(\mathbf{x}) := z = 0.023773K + 15\sum_{k=1}^{K} x_k, \tag{13}$$

which is just an affine function of the sum of all the deviation losses.

Next consider the good solution function f in (12). Player 1 is the decision maker, player 2 is the object of player 1's conjectures, and the arguments x_1 and x_2 are g-scores for player 1 and 2, respectively. Figure 1 plots f on a domain which includes a set of four points that represent the g-scores for a hypothetical 2×2 game. On that set, f attains its maximum at $(x_1, x_2) = (1, 1)$, and player 1 solves the game at this strategy profile (an action and a conjecture) which corresponds to that pair of g-scores.

Figure 1: Plot of the good solution function f in (12), embedding four pairs of (good reply) g-scores for a hypothetical 2×2 game.

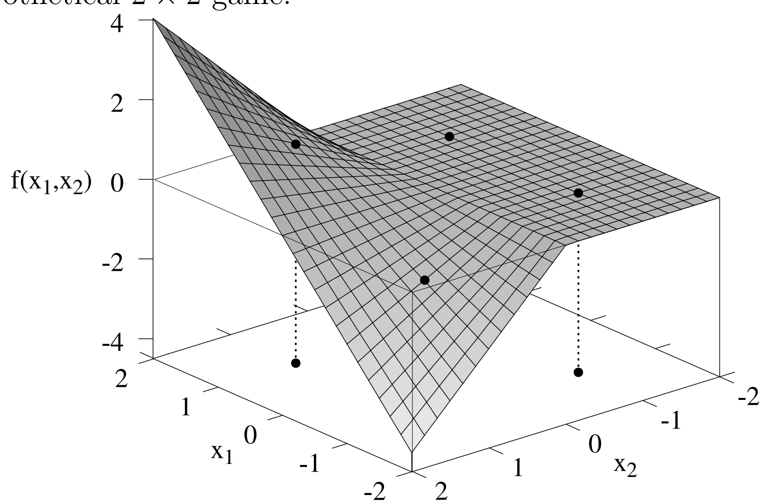


Figure 1 shows that f has a distinct, simple structure. From (12) one finds that for $x_2 > 0$, $f(x_1, x_2) = x_1 \cdot x_2$, and for $x_2 < 0$, f returns some number very close to 0. The best solutions have $(x_1, x_2) > 0$, i.e., where the player's action and conjecture are mutual good replies. Strict Nash equilibria have g-scores in this orthant, where f attains its maximum at some strategy profile that maximizes the product of the g-scores. This yields risk dominant solutions to 2×2-games, and offers/demands at about 40% of the total in ultimatum games.

The worst solutions have $x_1 < 0$ and $x_2 > 0$, with $f(x) < 0$. However, no game will be solved in this orthant because by (13), player 1 can choose any strategy profile in which he plays a best reply to obtain $x_1 > 0$ and hence, by (12), $f(x) > 0$.

We next investigate whether the program structure in (12) and (13) is representative for the aggregate solution concept. To address that question, we use the programs of all 200,000 agents to test specific benchmarks as explained below.

Structure of the good reply function. The good reply function uses the iteration function γ in (4) to compute $\mathbf{z}^k = \gamma(x_k, \mathbf{z}^{k-1}, d(k), K)$ for $k = 1, \ldots, K$, where $\mathbf{z}^0 = \mathbf{0}$; $\mathbf{x} = (x_1, \ldots, x_K)$ is a vector of deviation losses; and $d()$ is a dummy variable which is 1 for $k = 1$ and 0 otherwise. It then sets $g(\mathbf{x}) = z_1^K$.

A good reply function g is said to be *additive* if there are constants α and β such that the sequence $(z_1^1, \ldots, z_1^K)$ produced by γ satisfies $z_1^k - z_1^{k-1} = \alpha + \beta x$ for all $k = 2, ..., K-1$. The difference $z_1^1 - z_1^0$ is excluded from this definition because (1) z_1^0 is an arbitrary initial value, and (2) γ may be sensitive to the dummy variable $d()$, e.g., it may reset z_1^0 at the beginning of the first iteration.

The g-function in (13) is an additive good reply function. To measure the extent to which all g-functions have this property we proceed as follows: For each run of the model, randomly generate 100 data points $\{(x_1^n, x_2^n)\}_{n=1}^{100}$, each of which is a vector of $K = 2$ deviation losses. For each agent a and each data point (x_1^n, x_2^n), compute $\mathbf{z}^1 = \gamma^a(x_1^n, \mathbf{z}^0, d(1), K)$ and $\mathbf{z}^2 = \gamma^a(x_2^n, \mathbf{z}^1, d(2), K)$. The quantity $y_n^a := z_1^2 - z_1^1$ is an increase in a's good reply score which can be attributed to x_2^n. Regress $\{y_n^a\}_{n=1}^{100}$ on $\{x_2^n\}_{n=1}^{100}$ and and use R^2 as a measure of the degree to which a's good reply function is additive. Say that the run exhibits additive good replies if the mean R^2 across all agents in that run exceeds 0.99. Applying this criterion to all 100 runs, we find that all of them exhibit additive good replies. We can therefore conclude that additive good replies is a characteristic feature of the ASC.

Structure of the good solution function. Consider next the agents' good solution functions f. Unlike the good reply functions g, these functions are heterogeneous in such ways that do not seem to admit any simple characterization. We therefore attempt to extract a representative f-function as the pointwise median on a common discrete domain for all individual agents' good solution functions f^a. Since the arguments to f^a are g^a-scores, which are not comparable across agents, we consider instead the composite function $v^a : \mathbb{R}^2 \to \mathbb{R}$, defined as

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

The function v^a in (14) is a numerical representation of agent a's solution concept restricted to 2×2 games.

In (14), x_1 and x_2 are objective deviation losses for player 1 and player 2 respectively, and the affine function g^a translates these deviation losses to fit the subjective scale required by f^a. Since v^a and f^a only differ by an affine transformation g^a, we can aggregate (after normalization) the functions v^a across agents a to obtain the equivalent of an aggregate f-function. To that end, we introduce a fixed 3×3 grid of points $X^2 = X \times X$, where $X = \{-10, 0, 10\}$. For each $x \in X^2$ and each agent a, define

$$V^a(x) := \frac{v^a(x_1, x_2)}{v^a(10, 10)} \tag{15}$$

$$dV_1^a(x) := \frac{v^a(x_1 + 1, x_2) - v^a(x_1, x_2)}{v^a(10 + 1, 10) - v^a(10, 10)} \tag{16}$$

$$dV_2^a(x) := \frac{v^a(x_1, x_2 + 1) - v^a(x_1, x_2)}{v^a(10, 10 + 1) - v^a(10, 10)}. \tag{17}$$

The normalization by $v^a(10, 10)$ is intended to eliminate scale differences between the v-functions of different agents. We choose the point $(10, 10)$ as the basis for the normalization based on an expectation that each v^a will take on a positive value at that point.[10] The functions $V^a(x)$ in (15) can be compared across agents on the common domain X^2. Functions $dV_i^a(x)$ are normalized proxies for the partial derivatives of v^a as (16) and (17) measure the effect on V^a of a one unit increase in x_i from a point $x \in X^2$, normalized by the effect of

[10] To be precise, we expect that for most agents a, $v^a(x) = x_1 \cdot x_2$ (or some positive affine transformation thereof) on the positive orthant.

increasing x_i by one unit from point $(10, 10)$. Functions V^a, dV_1^a and dV_2^a are aggregated across all agents and runs by first taking pointwise medians in X^2 across all agents in each run, and then pointwise medians in X^2 across all runs. The three overall medians are denoted V, dV_1, dV_2, and their values on X^2 are shown in Table 19.

Table 19: Aggregate V-scores for 9 different combinations of deviation losses (x_1, x_2) for the two players. Player 1 is the decision maker and player 2 is the object of player 1's conjectures. V is a normalized numerical representation for the aggregate solution concept restricted to 2×2 games, and dV_i is a proxy for its normalized partial derivative with respect to x_i. The table is sorted in descending order with respect to $V(x)$.

Point	x_1	x_2	$V(x)$	$dV_1(x)$	$dV_2(x)$
1	10	10	1.0e+00	1.0e+00	1.0e+00
2	10	0	4.1e−05	3.2e−05	1.0e+00
3	0	10	3.2e−05	1.0e+00	3.2e−05
4	0	0	1.1e−09	5.1e−05	3.2e−05
5	10	−10	−9.1e−07	0.0e+00	1.4e−06
6	0	−10	−1.5e−05	0.0e+00	3.2e−10
7	−10	0	−1.7e−02	8.7e−03	−4.5e−01
8	−10	−10	−3.9e−02	1.2e−02	0.0e+00
9	−10	10	−1.0e+00	1.0e+00	−1.0e+00

We first check whether the function V in Table 19 agrees with the general shape of the specific good solution function f in Figure 1. To facilitate the comparison the results in Table 19 are sorted in descending order with respect to the normalized score $V(x)$. Consider first the points 1, 3 and 9 where $x_2 = 10$. As x_1 decreases from 10 to -10, V decreases from 1 to -1 while dV_1 is constant at 1, i.e., V is approximately linear in x_1 for $x_2 = 10$. In addition dV_2 decreases almost linearly from 1 to -1. These observations are consistent with $V(x) = x_1 \cdot x_2$ for $x_2 > 0$, which agrees with the function f in Figure 1. Second, consider the points 2 and 3, where (x_1, x_2) equals $(10, 0)$ and $(0, 10)$, respectively. On these points, V is close to zero and dV_1 and dV_2 are symmetric. These properties are also consistent with $V(x) = x_1 \cdot x_2$ for $x \geq 0$. Third, Table 19 shows that V is everywhere increasing in x_1, except for two asymmetric points (5 and 6) in player 2's disfavor, where $dV_1 = 0$. The corresponding asymmetric points in player 1's disfavor are 9 and 7, and on these points V

is actually decreasing in x_2. Thus the ASC seems to dislike asymmetric outcomes, and this dislike is stronger if the asymmetry is in player 1's disfavor. Fourth, on points where $x_2 \leq 0$, the function V takes on values close to 0 which is consistent with Figure 1.

We next consider the structure of function V when it takes on values close to 0. Table 19 reveals that V has an interesting structure on this subdomain. Consider the values of V in the descending order of Figure 19. Moving from the best point 1 to the next best ones (2 and 3), reduces the value of V by more than four orders of magnitude. Moving from points 2 and 3 to 4 reduces V by another four orders of magnitude. Similar, but less pronounced jumps occur for values further down in the table. This suggests that the solution concepts aim at creating a lexicographic ordering of their domains as follows: Start by looking for a strategy profile which yields a pair of positive g-scores. If there is none, look for a pair of non-negative scores, where one score, and preferably that of player 1, is positive. If there is still none, look for a pair of zero g-scores, and if none still exists, pick some pair where player 1 has a positive, or at least non-negative g-score (Table 19, points 5 and 6). Points 7, 8 and 9 will never be chosen because Player 1 (the decision maker) can always obtain $x_1 \geq 0$ by choosing a strategy profile where her action is a best reply to her conjecture. Consequently, an agent with this solution concept will never choose an action which is not a good reply to her conjecture. This is individual rationality for agents who think in terms of good replies.

The combination of this lexicographic structure for the good solution function with the additive structure for the good reply function implies that the ASC will first try to solve a game by maximizing the product of something which is close to the players' sums of deviation losses. This can be seen as a generalization of risk dominance from 2x2 games to all bimatrix games. Games with strict Nash equilibria and other games with positive g-scores can be solved in this way. If that is not possible, the ASC will look for strategy profiles that yield non-negative g-scores to both players, with a preference for larger g-scores, and breaking ties to the benefit of player 1 (the decision maker). Some games with weakly dominated strategies will be solved in this way, in which case the ASC will prefer more strategic stability for either player even if it is zero for the other player. If a game must be solved at an action pair with one negative g-score, the zero dV_1 and positive dV_2 at points 5 and 6 of Table 19 suggest that the ASC will choose a pair of pure strategies that maximizes the g-score to player 2 from the set of pairs that yield a non-negative g-score for player 1.

The last point is illustrated in Table 20 which depicts a game with no Nash equilibrium in pure strategies. The mixed strategy equilibrium is $(\frac{3}{4}A + \frac{1}{4}B, \frac{1}{2}A + \frac{1}{2}B)$. Suppose the additive good solution function is just the sum of its arguments, i.e. one deviation loss in the 2x2 game in Table 20. Then player 1 has positive g-scores at (B, A) and (A, B), which yield g-scores of -3 and -1, respectively, to player 2. This suggests that player 1 will solve the game at (A, B), and indeed, 74% of them do so. Player 2 has positive g-scores at (A, A) and (B, B), and both yield a g-score of -1 to player 1. This is consistent with the observation that the agents in position 2 solve the game at (A, A) and (B, B) with almost equal probabilities (44 and 49%).

Table 20: A game with no Nash equilibrium in pure strategies. Numbers in italics are probabilities (%).

(a) Payoffs, actions and conjectures

(s,t)		A	B	
	σ	*49*	*51*	Col
A	*79*	$1,1$	$1,0$	*47*
B	*21*	$2,1$	$0,4$	*53*
	Row	*23*	*77*	ϕ

(b) Solution, actions and conjectures

(s,t)		A	B	
	σ	*49*	*51*	Col
A	*79*	*5,44*	*74, 2*	*47*
B	*21*	*18, 4*	*2,49*	*53*
	Row	*23*	*77*	ϕ

In Section 3.1 we found that the ASC sometimes behaves as if the agents were concerned with social norms. Having identified the structure of the ASC we can now explain why.

The individual solution concepts resemble social welfare functions, where the good solution function aggregates two good reply scores, in some cases by maximizing their product. A social welfare function would work on payoffs instead of good reply scores, but there is a positive correlation between those two variables. In the Rock, Paper, Scissors game (Table 7), the correlation is 100% and in the Gift exchange game (Table 16) it is 68%. Due to that correlation, the ASC solves these games as if it tries to make a fair compromise in terms of payoffs, while in fact, it tries to balance the players' incentives to deviate. The solution to the Rock, Paper, Scissors game should therefore be reinterpreted as follows: 57% of the agents are equally concerned about their own and the opponent's incentives to deviate, 39% are more concerned with their own incentives, and 3% are more concerned about the opponent's incentives to deviate from a candidate solution.

3.4 Robustness checks

In this section, we check whether the ASC is robust with respect to two changes to the model specification. In the first one, we consider the algorithm which computes good replies and ask if initialization by zero values could have introduced a bias towards additive good replies. In the second, we test the effect of requiring that individual solution concepts be rationalizable. For 2x2 games rationalizability is equivalent to iterative elimination of strictly dominated strategies. This yields the four experiments shown in Table 21, where D0 is the base case that we have been considering so far.

Table 21: Robustness checks

Strictly dominated strategies	Memory initialization	
	Zero	Random
Allowed	D0	DR
Not allowed	N0	NR

3.4.1 Memory initialization

The algorithm in Section 2.1 which computes 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. An additive good reply function can then 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. To deliver an additive good reply function, the genetic programming algorithm must then produce some code which re-initializes the first memory slot to zero at the first iteration, but not on subsequent iterations. This is not likely to occur unless there is a fairly strong selection pressure in favor of additive good reply functions.

We do 100 runs of this experiment, called DR in Table 21, and find that 94 of those runs fall into two disjoint sets: The first set consists of 48 runs in which the agents have additive good reply functions. The second set consists of 46 runs in which the frequency of Nash play is at least 99% across all games with one pure Nash equilibrium. By comparison, the mean frequency of Nash play in such games for the D0 experiment was only 83.8%, see Table 17.

A representative agent from the second set of 46 runs will be called a *Nash player*. By analyzing some of those agents by means of the simplification technique explained in footnote 9, we find that their good reply functions have a multiplicative structure. To check if this is representative for all Nash players, we proceed by analogy with the test of additive good replies in Section 3.3: For each run, randomly generate 100 data points $\{(x_1^n, x_2^n)\}_{n=1}^{100}$, each of which is a vector of $K = 2$ positive payoff differences. For each agent a and data point (x_1^n, x_2^n), compute $\mathbf{z}^1 = \gamma^a(x_1^n, \mathbf{z}^0, d(1), K)$, $\mathbf{z}^2 = \gamma^a(x_2^n, \mathbf{z}^1, d(2), K)$, where, as before, the dummy variable $d(k)$ is 1 for $k = 1$ and 0 otherwise. The quantity $y_n^a := z_1^2/z_1^1$ is a relative increase in a's good reply score which can be attributed to the positive payoff difference x_2^n. Regress $\{\log(y_n^a)\}_{n=1}^{100}$ on $\{\log(x_2^n)\}_{n=1}^{100}$ and use R^2 as a measure of the degree to which a's good reply function is multiplicative.

The correlation matrix in Table 22 shows that Nash play and multiplicative good replies are closely related, and that these two variables are negatively correlated with additive good replies. Also, by sorting the 100 DR runs by increasing Nash play score, all runs with additive $R^2 > 0.99$ end up in the lower 50 runs, and all runs with multiplicative $R^2 > 0.99$ end up in the upper 50 runs in the sorted set of runs.

The functions γ and f for a representative Nash player are listed in (18) and (19). Both functions have been simplified as explained in footnote 9. Consider first the good reply

Table 22: Correlation matrix for R^2's from tests of *additive* and *multiplicative* good replies, and the frequency of Nash play in games with one pure Nash equilibrium (*playNash*). For each variable, the number of observations is 100.

	Additive	Multiplicative	playNash
Additive	1.00		
Multiplicative	-0.90	1.00	
playNash	-0.95	0.90	1.00

iteration function γ in (18). On entry to this function, the memory slot z is initialized to some random number, but γ resets it to 1 at the first iteration ($k = 1$). Then, for each iteration $k = 1, 2, \ldots, K$, it computes $z \leftarrow \gamma(x_k, z, k)$, where, as before, each deviation loss x_k is the difference between two standard normal variates, each scaled by 10, rounded to the nearest integer, and then scaled by some random real number in the interval $[0.01, 100]$.

$$
\gamma(x_k, z, k) = \begin{cases} 3.69 \cdot 10^{13} \cdot (x_k + 1.74 \cdot 10^{-4}) \cdot \max(0, 1) & \text{if } k = 1 \\ 3.69 \cdot 10^{13} \cdot (x_k + 1.74 \cdot 10^{-4}) \cdot \max(0, z) & \text{if } k > 1. \end{cases} \tag{18}
$$

$$
f(z_1, z_2) = \max(z_1, \max(0, z_1) \cdot z_2) \tag{19}
$$

When all deviation losses have been processed, the g-score is obtained as $g(\mathbf{x}) = z$.

The function γ produces positive g-scores if and only if all deviation losses are non-negative. The 'if' part is straightforward as z is then non-negative after the first iteration and non-decreasing under further iterations of (18) if all $x_k \geq 0$. The 'only if' part can be seen as follows. Since, by construction, $\text{abs}(x_k) \geq 10^{-2}$ if $x_k \neq 0$, we have $\text{sgn}(x_k + 1.74 \cdot 10^{-4})$ $= \text{sgn}(x_k)$ for $x_k \neq 0$. Thus a negative deviation loss either results in a negative number z (if it is the last iteration) or in a zero (as z is set to zero in the next iteration and remains there due to the term $\max(0, z)$). Thus the random order in which deviation losses are presented to γ can lead to a negative or a zero score if one or more deviation losses are strictly negative. But it cannot produce a positive score. Observe that the agent still behaves in a consistent manner because the good solution function f in (19) does not distinguish between zero and negative arguments, as can be seen from the representation of f in Table 23.

Table 23: The good solution function $f(z_1, z_2) = \max(z_1, \max(0, z_1) \cdot z_2)$

	$z_2 \leq 0$	$z_2 > 0$
$z_1 \leq 0$	0	0
$z_1 > 0$	z_1	$z_1 \cdot z_2$

The entries in Table 23 follow directly from (19) except for the lower right cell which states that $f(z_1, z_2) = z_1 \cdot z_2$ if $(z_1, z_2) > 0$. But this holds because (18) implies that any strictly positive z_i generated by the iteration of γ actually satisfies $z_i > 1$.

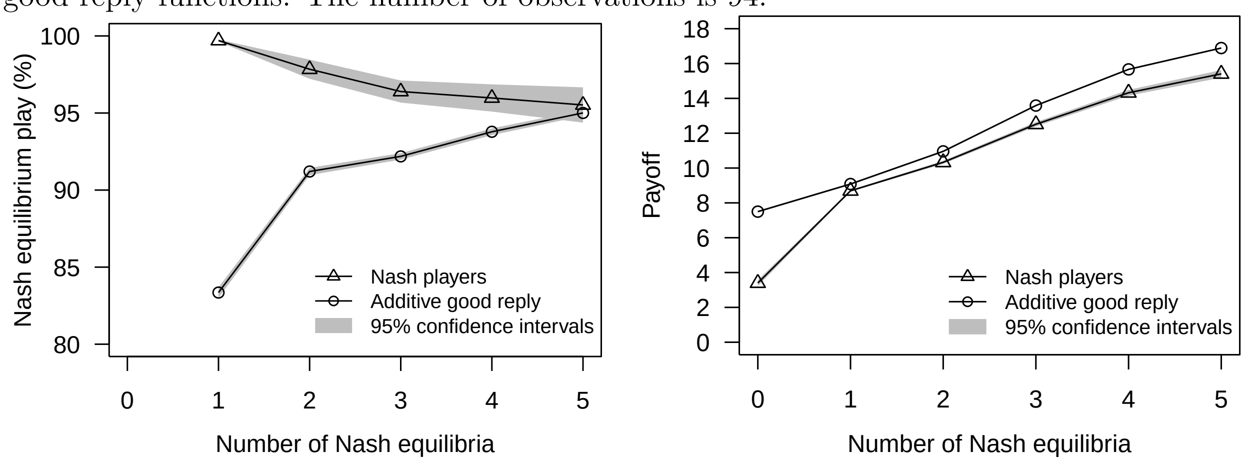
The solution concept defined by (18) and (19) has a number of appealing properties on games with one or more pure Nash equilibria. First, it selects one among those equilibria, and in this sense, it is a refinement of the Nash equilibrium concept. Second, on 2×2 games with two strict pure Nash equilibria it coincides with the Harsanyi-Selten risk dominance concept. Third, it features a multiplicative good reply function which generalizes the risk dominance concept to larger games and games with non-strict Nash equilibria. And fourth, this multiplicative structure parallels the additive one of the complementary 48 runs of this experiment.

The multiplicative solution concept loses its intuitive appeal when applied to games without pure Nash equilibria. Such games have no action profile where both good reply scores are positive. They will therefore be solved at the lower left cell of Table 23, namely at a strategy profile which maximizes the good reply score for player 1 (the decision maker). In other words, on games with no pure Nash equilibria, the multiplicative solution concept minimizes the decision maker's incentives to deviate without regard to the incentives of the opponent. This contrasts with the additive solution concept, which attempts to balance those incentives. It suggests that the multiplicative solution concept will perform worse than the additive one on games without pure Nash equilibria because its conjectures will be wrong more often.

Applying the multiplicative solution concept to the games in Section 3.1, we find less cooperation and lower aggregate payoffs as compared to the additive solution one: The 'refinement games' in Tables 10 and 11 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.

As a last step in the analysis of the multiplicative solution concept we compare its behavior with that of its additive counterpart across 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. We then solve each one of those 6,000 games for each agent in the 46 populations of Nash players and for each agent in the 48 populations with additive good reply functions. The results are plotted in Figure 2.

Figure 2: Behavior in experiment DR in games with a varying number of pure Nash equilibria by Nash players (agents with multiplicative good reply functions) and agents with additive good reply functions. The number of observations is 94.



The left panel of Figure 2 shows the frequency of Nash equilibrium play.[11] In games with one pure Nash equilibrium, the Nash players 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%. Agents with additive good reply functions are not equipped to identify Nash equilibria. Instead they look for strategy profiles with positive sums of devi-

[11] 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 $1/n$.

ation 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 subpopulations with respect to the frequency of Nash equilibrium play.

The right panel of Figure 2 plots payoffs against the number of pure Nash equilibria for the two subpopulations. 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 Nash players fare particularly badly in games with no pure Nash equilibrium, obtaining less than half the payoff of the additive agents. This is consistent with our previous observation that agents with multiplicative solution concepts tend to base their actions on false conjectures in games without pure Nash equilibria.

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 includes strictly dominated strategies. While strictly dominated solutions agree with intuition or experiments for some games, it raises the issue of whether the solution concept is somehow robust with respect to addition of dominated strategies.[12] To illustrate the problem, we consider the game in Table 24.

The symmetric game in Table 24 has one Nash equilibrium (A, A) in pure strategies, with payoffs $(1, 1)$. The ASC (with additive good replies) 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) is selected because it has a high g-score of $9 = (10-11)+(10-0)$. The

[12]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.

Table 24: 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

small negative term $(10 - 11)$ comes from weak incentives to deviate, but the large positive term $(10 - 0)$ is due to the presence of the dominated action c. So even if the ASC may 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 (as seen in Section 3.1) is solved at (A, A).

It is easy to construct this type of examples by adding strictly dominated strategies to an existing game. The additive good reply functions are vulnerable to such transformations because they do not distinguish between more and less relevant alternatives. 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 24 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 evolution. We will therefore impose IESDS ex ante and see if and how this affects the aggregate solution concept.

To that end, we do 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 21. This restriction is imposed by iteratively removing all strictly dominated strategies from any game before applying some solution concept.

We first do 100 runs with experiment NR (IESDS and random initial memory). This yields 81 runs with additive good replies and only one run with Nash players. Similarly, 100 runs with experiment N0 (IESDS and zero initial memory) yields 93 runs with additive good replies and no runs with Nash players. Thus IESDS strengthens the additive solution concept at the cost of the multiplicative one. As discussed above, IESDS removes some potentially irrelevant alternatives from games with strictly dominated strategies, which might benefit the additive solution concept. Apparently, this effect is strong enough, or the competition from multiplicative Nash players is weak enough for the additive solution concept to achieve dominance when IESDS is imposed.

4 Conclusion

The paper uses a genetic programming algorithm to study evolution of initial play in bimatrix games. The model has 2,000 artificial agents who gain experience with initial 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 something akin to a coarse lexicographic ordering 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 2x2 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 artifacts due to positive correlation between payoffs and deviation losses, and a solution concept which resembles a social welfare function, e.g, 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 initial 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) By representing games by means 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. (3) 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. (4) The model can be modified to take a closer look at equilibrium refinements. In one of our model variants, we imposed rationalizability by iteratively eliminating all strictly dominated strategies before applying a solution concept. Similarly, one could compute all mixed Nash equilibria and eliminate all strategies that are not in the support of any such equilibrium before applying a solution concept. Solution concepts evolved on such a restricted domain of games would be more tailored to the task of equilibrium refinement and might be able to beat the one studied here at that task.

References

Aumann, R. J. (1974). Subjectivity and correlation in randomized strategies, *Journal of Mathematical Economics* **1**(1): 67–96.

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.

Binmore, K. (1987). Modeling rational players: Part I, *Economics & Philosophy* **3**(2): 179–214.

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* A. K. K. Binmore & P. Tani (eds), *Frontiers of Game Theory*, Addison-Wesley, Reading, MA, pp. 237–254.

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*, PIER Working Paper No. 18-009.

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.

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.

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.

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

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.

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

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.

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. (1998). Features of experimentally observed bounded rationality, *European Economic Review* **42**(3-5): 413–436.

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.

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

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.