Evolution of Strategies in Repeated Stochastic Games

Anders Eriksson
Kristian Lindgren

SFI WORKING PAPER: 2001-04-023

SFI Working Papers contain accounts of scientific work of the author(s) and do not necessarily represent the views of the Santa Fe Institute. We accept papers intended for publication in peer-reviewed journals or proceedings volumes, but not papers that have already appeared in print. Except for papers by our external faculty, papers must be based on work done at SFI, inspired by an invited visit to or collaboration at SFI, or funded by an SFI grant.

©NOTICE: This working paper is included by permission of the contributing author(s) as a means to ensure timely distribution of the scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the author(s). It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may be reposted only with the explicit permission of the copyright holder.

www.santafe.edu
Evolution of strategies in repeated stochastic games with full information of the payoff matrix

Anders Eriksson and Kristian Lindgren
Department of Physical Resource Theory, Complex Systems Group Chalmers University of Technology and Göteborg University SE-41296 Göteborg, Sweden
Corresponding author: K. Lindgren (frtkl@fy.chalmers.se, phone: +46(0)31 - 772 3131)

Abstract
A framework for studying the evolution of cooperative behaviour, using evolution of finite state strategies, is presented. The interaction between agents is modelled by a repeated game with random observable payoffs. The agents are thus faced with a more complex (and general) situation, compared to the Prisoner’s Dilemma that has been widely used for investigating the conditions for cooperation in evolving populations. Still, there is a robust cooperating strategy that usually evolves in a population of agents. In the cooperative mode, this strategy selects an action that allows for maximizing the payoff sum of both players in each round, regardless of the own payoff. Two such strategies maximize the expected total payoff. If the opponent deviates from this scheme, the strategy invokes a punishment action, which for example could be to aim for the single round Nash equilibrium for the rest of the (possibly infinitely) repeated game. The introduction of mistakes to game actually pushes evolution to more cooperative, even though at first sight, it makes the game more cooperative.

1 Introduction

The conditions for cooperative behaviour to evolve have been the focus in a large number of studies using genetic programming, evolutionary algorithms, and evolutionary modelling (Matsuo 1985, Axelrod 1987, Miller 1989, Lindgren 1992, 1997, Nowak and May 1993, Stanley et al 1993, Lindgren and Nordahl 1994, Ikegami 1994, Nowak et al 1995, Wu and Axelrod 1995). Most of these studies have used the iterated Prisoner’s Dilemma (PD) game as a model for the interactions between pairs of individuals (Rapoport and Chammah 1965, Axelrod 1984, Sugden 1986), and in the PD game cooperation is the action that lets two players share the highest total payoff. The problem in the single round PD is that one is tempted to defect to get a higher score, but when the game is repeated one usually finds that various types of cooperative behaviour evolves, for example in the form of the Tit-for-tat strategy (Axelrod 1984).

The basis for cooperation to emerge in these models is the repeated encounters between players. In “real” situations one usually meets the other player again,
but seldom in an identical situation. Therefore, we suggest and analyse a more open game-theoretic problem as a basis for the interactions between players. In each round a completely random payoff matrix is generated, and the players observe the payoff matrix before choosing their actions. In order to compare this model with the iterated PD, we introduce a parameter that can continuously change the game from the iterated PD to the repeated random payoff game.

In this model cooperation is not associated with a certain action, but rather with how one chooses to act depending on the structure of the present payoff matrix in combination with how the opponent has acted in previous rounds. There is a very strong cooperative strategy in this type of game, namely the strategy that always plays the action that aims for the highest payoff sum for the two players, regardless of the own score. If two such players meet they will, in the long run, get equal scores at the maximum possible level. The circumstances for such cooperative strategies to evolve may be critical, though, and this paper is a starting point for investigating what strategy types evolve in this type of open game. It is also demonstrated that when mistakes are introduced, evolution leads to strategies that in a more subtle way keeps the high level of cooperation, even if the game with mistakes appears to be more tricky.

The model presented here is a population dynamics model with strategies interacting according to their abundances in the population. The equations of motion for the different variables (the number of individuals) are thus coupled, resulting in a coevolutionary system. The success or failure of a certain type of individual (species) depends on which other individuals are present, and there is not a fixed fitness landscape that is explored by the evolutionary dynamics.

The paper is organised as follows. First we give a brief presentation of the repeated stochastic game and some of its characteristics. Then we describe the agents including the strategy representation, and how the population dynamics with mutations lead to a coevolutionary model. In the final sections we introduce mistakes to the model, and briefly discuss some preliminary results and directions for future research within this model framework.

## 2 Random payoff game

The repeated random payoff game used in this study is a two-person game in which each round is characterized by two possible actions per player and a randomly generated but observable payoff matrix. The payoff matrix elements are random, independent, and uniformly distributed real numbers in the range $[0, 1]$. New payoffs are drawn every round of the game, as in Table 1.

The agents playing the game have complete knowledge of the payoffs $u_{jk}(t)$ for both players $i \in \{A, B\}$ and possible actions $j$ and $k$ for the present round $t$. Here all $u_{jk}(t)$ are independent random variables with uniform distribution.

The single round game can be characterised by its Nash equilibria (NE), i.e., a pair of actions such that if only one of the players switches action that player
will reduce her payoff. If
\[(u_A^{00} - u_A^{10})(u_A^{01} - u_A^{11}) > 0\]
or
\[(u_B^{00} - u_B^{01})(u_B^{10} - u_B^{11}) > 0,\] (1)

there is exactly one (pure strategy) Nash equilibrium in the single round game. This occurs in 3/4 of the rounds. If this does not hold, and if
\[(u_A^{00} - u_A^{10})(u_B^{00} - u_B^{01}) > 0,\] (2)

there are two (pure strategy) NEs, which occurs in 1/8 of the rounds. Otherwise, there are no (pure strategy) NEs.

There are a number of simple single round (elementary) strategies that are of interest in characterising the game. Assume first that there is exactly one NE, and that rational (single round) players play the corresponding actions. The payoff in this case is \(\max(x, h)\), where \(x\) and \(h\) are independent uniformly distributed stochastic variables in \([0, 1]\), and this results in an expectation value of 2/3 \(\approx 0.667\).

Let us define a strategy “NashSeek” as follows. If there is only one NE in the current payoff matrix, one chooses the corresponding action. If there are two NE, one aim for the one that has the highest sum of the two players payoffs, while if there is no NE, one optimistically chooses the action that could possibly lead to the highest own payoff.

A second strategy, “MaxCoop”, denoted by \(C\), aims for the highest sum of both players’ payoffs. If two such players meet, they score \(\max(x_1 + h_1, x_2 + h_2, x_3 + h_3, x_4 + h_4)\) together, where \(x_i\) and \(h_i\) are independent uniformly distributed stochastic variables in \([0, 1]\), and this results in an expectation value of \(s_C = 3589/5040 \approx 0.712\).

If a player using MaxCoop wants to avoid exploitation, some punishment mechanism must be added. A new elementary strategy could be invoked for this, so that if the opponent deviates from MaxCoop, then the defector will at the most get the minmax score for the rest of the game. We call such an elementary strategy “Punish”, denoted by \(P\), and the strategy being punished will at the most get \(\min(\max(x_1, x_2), \max(x_3, x_4))\), with an expectation value of \(s_P = 8/15 \approx 0.533\). Other forms of punishment, that are less costly for both players, are also possible. In Table 2 we show the expected payoffs between the simple strategies in this paper. All expected payoffs for the simple strategies were computed analytically.

This illustrates that both players choosing MaxCoop with punishment correspond to a Nash equilibrium in the infinitely (undiscounted) iterated game. In a discounted iterated game, with a probability \(\delta\) for the game to end in each round, the condition for avoiding exploitation is \(1 + (1 - \delta)s_P/\delta < s_C/\delta\). This means that if \(\delta < (s_C - s_P)/(1 - s_P) = 901/2352 \approx 0.383\), then both choosing MaxCoop with punishment is a NE.

\(^1\)We assume that no payoff values are identical, so that Nash equilibria are strict. Rounds for which this does not hold are of measure zero.
Table 1: Payoff matrix for the stochastic game depending on the parameter $r$ that brings the game from an ordinary PD to the fully stochastic game. The symbols $\eta_i$ denote independent uniformly distributed random variables in the interval $[0, 1]$, and note that for $r = 1$ these form the payoffs of the game.

<table>
<thead>
<tr>
<th>Action 0</th>
<th>Action 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action 0</td>
<td>$(1 - r) + r\eta_1$</td>
</tr>
<tr>
<td></td>
<td>$(1 - r) + r\eta_5$</td>
</tr>
<tr>
<td>Action 1</td>
<td>$\frac{2}{3}(1 - r) + r\eta_7$</td>
</tr>
</tbody>
</table>

Table 2: Expected payoffs for the simple strategies, for the row player. The strategies are defined in section 3.1.

<table>
<thead>
<tr>
<th>MaxMax</th>
<th>Punish</th>
<th>MaxCoop</th>
<th>NashSeek</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxMax</td>
<td>0.600</td>
<td>0.400</td>
<td>0.710</td>
</tr>
<tr>
<td>Punish</td>
<td>0.500</td>
<td>0.400</td>
<td>0.549</td>
</tr>
<tr>
<td>MaxCoop</td>
<td>0.592</td>
<td>0.431</td>
<td>0.712</td>
</tr>
<tr>
<td>NashSeek</td>
<td>0.643</td>
<td>0.443</td>
<td>0.750</td>
</tr>
</tbody>
</table>

In order to be able to illustrate how this model is an extension of the iterated Prisoner's Dilemma game, we introduce a stochasticity parameter $r$ ($0 = r = 1$) that changes the payoff characteristics from the simple PD game to the fully stochastic payoff. The $r$-dependent payoff matrix is shown in Table 1.

### 3 Evolutionary model

We consider a population of $N$ agents, competing for the same resources. The population is at the limit of the carrying capacity level of the environment, so the number of agents $N$ is fixed.

In each generation, the agents play the repeated stochastic payoff game with the other agents, and reproduce to form the next generation. The score $s_i$ for strategy $i$ is compared to the average score of the population, and those above average will get more offspring, and thus a larger share in the next generation. Let $g_{ij}$ be the score of strategy $i$ against strategy $j$, and let $x_i$ be the fraction of the population occupied by strategy $i$. The score $s_i$ is then

$$s_i = \sum_j g_{ij}x_j,$$  \hspace{1cm} (3)

and the average score $s$ is

$$s = \sum_i s_ix_i.$$  \hspace{1cm} (4)
We define the fitness \( w_i \) of an individual as the difference between its own score and the average score:

\[
w_i = s_i - s
\]  

(5)

Note that this ensures that the average fitness is zero, and thus the number of agents is conserved. The fraction \( x_i \) of the population of the strategy \( i \) changes as

\[
x_i(t+1) = x_i(t) + d \cdot w_i \cdot x_i(t) = x_i(t) + d (s_i - s) \cdot x_i(t),
\]  

(6)

where \( d \) is a growth parameter, and all terms to the right of the first equality sign are evaluated at time \( t \).

When an individual reproduces, a mutation may occur with low probability. The mutations are described in detail in section 3.2. In this way, diversity is introduced in the population. In the population dynamics, this corresponds to a number of mutations per strategy proportional to the strategies share of the population. The share of the mutant strategy is increased by \( 1/N \) (corresponding to one individual), and the mutated strategy loses the same share.

### 3.1 Players and strategies

In many approaches to evolutionary game-theoretic models, strategies are represented as deterministic finite state automata (FSA) (Miller 1988, Nowak et al 1995, Lindgren 1997), where each transition conditions on the action of the opponent. Each state is associated with an action, and a certain state is the starting state. In order to calculate the payoffs for a pair of players, an FSA describing the joint states of the players is constructed. The expected undiscounted payoffs are then computed from this FSA.

However, since our games are stochastic, the result of each action varies between rounds. In this case, a natural extension is to act according to a simple behaviour rather than a pure action. In this article, we choose the behaviours to be deterministic maps from a payoff matrix to an action, and we call these elementary strategies. In our first experiments, the behaviours available to the agents are MaxMax (M), Punish (P), MaxCoop (C), and NashSeek (N), described as follows:

- **MaxMax**: Select the action so that it is possible to get the maximum payoff for the agent in the payoff matrix. Greedy and optimistic.
- **Punish**: Select the action that minimizes the opponent’s maximum payoff. Note that this may be even more costly for the punishing strategy.
- **MaxCoop**: Assume that the other player also plays MaxCoop, and select the action that maximizes the sum of the players’ payoffs.
- **NashSeek**: If there is only one Nash equilibrium, select the corresponding action. If there are two Nash equilibria, select the one that gives the highest sum of the players’ payoffs. If there are no equilibria, play according to MaxMax (see above).
Agents thus have four possible elementary strategies, each determining what specific action should be chosen when a certain payoff matrix is observed. We choose to have only four simple strategies, since we want to keep our system as simple as possible. The MaxCoop is a natural member, since it is the optimal behaviour at self-play. In our choices of the other behaviours, we have tried to find the most interesting aspects of the game: Nash equilibria, punishment against defection, and greediness.

Since the agents cannot access the internal state of the opponent, but only observes the pure actions, we build an action profile by comparing the opponent’s action to the actions that the opponent would have taken, given that it followed each of the elementary strategies. In this way, a player may determine whether the opponent’s action is consistent with a certain elementary strategy.

We now construct an FSA strategy representation that allows for more complex (composite) strategies by associating different internal states with (possibly) different elementary strategies. In each state, the agent has a list of transitions to states (including the current state). Each transition is characterized by a condition on the opponent’s action profile. For each elementary strategy, the agent may require a match or a non-match, or it may be indifferent to that behaviour. The transitions are matched in the order they occur in the list, and the first matching transition is followed. If no transition matches, the state is retained to the next step.

When solving for the players' payoffs, we construct the joint state FSA of the whole iterated game (a more detailed description is in Appendix A). The expected payoffs from all pairs of behaviours, and the probability distributions over the set of action profiles are precomputed. Since the payoff matrices are independent stochastic variables, we can compute the probability distributions of the transitions from each node in the game FSA by summing over the action profile distribution. It is then straightforward to compute the resulting stationary distribution, and the corresponding payoffs.

3.2 Mutations

There are several different ways to change a strategy by mutations. Since we represent our strategies by graphs, there are some basic graph changes that are suitable as mutations.

A graph may be mutated by changing the number of nodes, changing the connectivity of the graph, or by changing the labels of the nodes. See Fig. 2 for an example mutation. Removing a node constitutes deleting the node, and removing all edges pointing to that node. Adding a node is made in such a way that the strategy is unchanged, by splitting an existing node as follows. A new node is created, and all the edges leaving the original node are copied to this node. All self-referential edges of the original node are redirected to the new node. Finally, a match-all edge to the new node is added last in the rule list of the original node.

We think that it is important that a strategy may change in size without changing the behaviour. If a strategy is at a fitness peak, many mutations that change the fitness may be needed to reach another peak, using only the mutations that
Figure 1: Example of a graph playing strategy. The letters in the nodes stand for the elementary strategies MaxCoop (C) and NashSeek (N), and the black ring indicates the initial state. The transition denoted **T** applies only when the opponent’s action is consistent with MaxCoop (the third elementary strategy), but does not care about the other elementary strategies. The transition rule **F** is used when the opponent does not follow the MaxCoop strategy. This composed strategy is similar to the Tit-for-tat strategy in that it cooperates by playing MaxCoop if the opponent played according to MaxCoop in the last round, otherwise it defects by playing NashSeek.

change the strategy. But if a strategy may grow more complex without endanger the good fitness, these mutation might bring it close to other peaks (in terms of using the other mutations). That genetic diversification among agents with identical fitness is important has been observed for example in hill climbing algorithms for satisfiability (SAT) problems (Selman et al 1992).

Since large random changes in a strategy often are deleterious, this allows the strategy to grow more complex in a more controlled way. For example, it might lead to a specialization of the old strategy.

The connectivity of the graph is mutated by adding a random rule, removing a rule, or by mutating a rule. Rules are mutated by changing the matching criterion at one location, or by changing the rule order (since the rules are matched in a specific order, this changes the probability of a match by that rule). Finally, a node may be mutated by randomly selecting a different strategy as a label. After each mutation, we perform a graph simplification where we remove all nodes that are unreachable from the start node, in order to keep the representation as compact as possible.

In our experiments, the most common mutation is the node strategy mutation. We keep the node addition and removal equiprobable, so that there is no inherent preference for strategy growth in the simulation. The same holds for rule addition and removal. All growth of the strategies that is observed in the population is thus due to the competition between strategies, and the need for complexity that arises as a consequence of that competition.

4 Results

The model described in the previous sections has been simulated and analysed in a number of ways. First, we have made a simple analysis of how the scores
between pairs of elementary strategies depend on the degree of stochasticity in the payoff matrix, in the range from the pure PD game to the fully stochastic payoff game. In Table 2, the results for a number of strategies are shown. The table shows clearly that there is a possibility for a cooperative behaviour scoring higher than the short term Nash equilibrium level.

The strategy NashSeekOpt (play optimally against NashSeek) is identical to the basic Nash-seeking strategy (NashSeek) when there is only one Nash equilibrium in the single round game, but plays optimal against NashSeek in case of no or two Nash equilibria (and thus exploiting NashSeek).

A player that wishes punish an opponent, may try to minimize the score for the opponent by playing a MinMax strategy, and the opponent’s score in that case is the second lowest curve in Fig. 3. From the figure we see that when the game becomes more stochastic there may be several elementary strategies that can work as a punishment, and that this allows for punishments of different magnitudes. In the PD game, of course, the only way to punish the opponent (in a single round) is to defect (which is equivalent to NashSeek and Punish).

In Fig. 4, we show the evolution of the average score in the population in the case of a fully stochastic payoff matrix. The score quickly (in a few thousand generations) increases to the level of the single round Nash equilibrium, but with sharp drops when mutant strategies manages to take a more substantial part of the population. After about 20,000 generations, the population is close to establishing a MaxCoop behaviour, but fails, probably because of the lack of a strong enough punishment mechanism. A new attempt is made after 60,000 generations, and here it seems that a composite strategy using MaxCoop against other cooperative strategies, but punishing exploiting opponents, is established.

The noisy behaviour observed in Fig. 4 is an effect of the sub-population of mutants created by the relatively high mutation rate (5%). There may be two mechanisms involved here. The immediate effect is that many mutants deviate from the MaxCoop strategy which induces all playing against them to switch to punishment, e.g., NashSeek. A secondary effect is that mutations that
Figure 3: By the stochasticity parameter $r$, we can make a continuous transition from a pure Prisoner’s Dilemma game to the fully stochastic payoff matrix. Here we show the first player’s score for different pairs of strategies as functions of $r$. The strategy PunishOpt plays optimally against Punish.

remove the punishment mechanism may survive for a while, growing in size by genetic drift, until a mutant that exploits its lack of punishment enters. Such an exploiting mutant may increase its fraction of the population at the cost of reducing both average population score and the size of the exploited strategy, eventually leading to the extinction of both of the mutants.

When the population reaches the score level of the MaxCoop strategy, there are periods when the score drops drastically down to the Nash level or even lower. A reasonable explanation is that the punishment mechanism that is necessary in order to establish the MaxCoop strategies when Nash-seeking strategies are present, is not needed when the Nash-seeking strategies have gone extinct. By genetic drift, MaxCoop strategies without a punishment mechanism may increase their fraction of the population, and then when a mutant NashSeek enters there is plenty of strategies for it to exploit, leading to a drop in average score. But then the MaxCoop with punishment can start to grow again, resulting in the noisy average score curve of Fig. 4. Occasionally, if the simulation is continued, there may be transitions back to the score level of the single round Nash equilibrium.
Figure 4: The average score in the population is shown for the first 1.4 million generations in the case of a fully stochastic payoff matrix. Population size is 100 strategies, the growth parameter $d$ is 0.1, and the mutation rate $m$ is 0.5. The score is normalized to the MaxCoop level. The thick lines indicate the normalized levels of MaxCoop vs. MaxCoop (upper) and NashSeek vs. NashSeek (lower). It is clear that the population finds a strategy with MaxCoop, that is resistant against greedy strategies for many generations.

5 Introducing mistakes

We model mistakes as a probability $\rho$ of taking the action opposite to the intended one, in each round and for all players. Since we consider infinite games, even a very small chance of mistakes has a huge impact on the overall payoffs and how the strategies evolve, even when the effect on the expected payoffs in the single round games are negligible (this effect is approximately proportional to $\rho$). As seen in (Lindgren 1992), strategies that punish deviations from a mutual elementary strategy must now take the chance of mistakes into account. A strategy like the one in Fig. 1, for example, will perform much worse since it will be lured into playing NashSeek much more often. Especially when playing against itself, it will inevitably end up with long sequences of mutual “defects” in that both players play NashSeek, and they will only get back on the cooperating track again (playing MaxCoop) if a new mistake occurs. It is thus essential that the players have some way of recovering from unintentional actions, from both parties. To this end, the players now monitor their own actions, as well as the opponents. A player still cannot detect when the opponent makes a mistake, but by “apologising” in some appropriate way (e.g. by playing MaxCoop) the
Figure 5: The average score in the population is shown for the first 2.6 million generations in the case of a fully stochastic payoff matrix, and $\rho = 0.01$. Population size is 100 strategies, the growth parameter $d$ is 0.1, and the mutation rate $m$ is 0.5. The score is normalized to the MaxCoop level. The thick lines indicate the normalized levels of MaxCoop vs. MaxCoop (upper) and NashSeek vs. NashSeek (lower). It is clear that the population finds a strategy with MaxCoop, that is resistant against greedy strategies for many generations.

long run cooperation might be preserved. But this mechanism may leave the player open for greedy players to make use of it, so it is more complicated to find an appropriate mechanism.

When there are no mistakes, the population often consists of players playing MaxCoop with some punishment for deviations, as can be seen in Fig. 4. When there is a large fraction of these agents, mutants from them without a punishment mechanism may still enter the population and grow by collecting mutants. When $\rho$ is finite, a strategy without an appropriate mechanism for error correction will not be able to enter the population. Even when the mutant has an error correction mechanism that works when playing against itself, it is unlikely that it will work as well when playing against another strategy. Fig. 5 shows a typical run with mistakes. Although the period of cooperation appears later than it usually does without mistakes (about 200,000 generations later), when it appears it is much more stable against invasion.
6 Discussion

It is clear from the payoff matrix describing the elementary strategies, that the iterated game (with a sufficiently low discount rate) has Nash equilibria in the form of both players using a certain form of MaxCoop-punishment, the most common one being MaxCoop with NashSeek as punishment.

But it is also clear that there are several strategies that are playing on equal terms with such a MaxCoop-punishment strategy. For example, strategies that by mutation loose their punishment mechanism may enter and increase their fraction of the population by genetic drift. This in turn leads to a population that is vulnerable to mutants exploiting the cooperative character of the MaxCoop behaviour, for example by the NashSeek behaviour. Thus, the Nash equilibrium that characterises the population dominated by MaxCoop-punishment is not an evolutionarily stable one, as can be seen in the simulations. The effect is the same as the one in the iterated Prisoner’s Dilemma that makes Tit-for-tat an evolutionarily non-stable strategy (cf. IPD; see, for example, (Boyd & Lorberbaum 1987, Boyd 1989)).

This genetic drift appears to be prevented by the introduction of a finite probability of mistakes in the actions of the elementary strategies. The strategies cooperating at a high payoff level have evolved mechanisms that can simultaneously protect the players from greedy strategies, and lead the game back to cooperation when a mistake occurs. We believe that this is the cause for the evolutionary stability of the system, since it is likely that a random mutation will destroy these mechanisms, and it is likely that different mechanisms doesn’t work as well together as they work with themselves.

The evolution of a population dominated by MaxCoop (with some punishment mechanism) is not unexpected. A population of players (using identical strategies) can always enforce a certain score $s^*$, if this score is larger than (or equal to) the smallest punishment score $s_P$ and smaller than (or equal to) the maximum cooperation score $s_C$, $s_P \leq s^* \leq s_C$. This means that the population can punish a new strategy (mutant) that enter, unless it adopts the same strategy and plays so that the score $s^*$ is reached in the long run. This argument follows the idea behind the Folk Theorem that appears in various forms in the game-theoretic literature (Fudenberg and Tirole 1991, Dutta 1995).

The repeated game with stochastic observable payoffs offers a simple model world in which questions on the evolution of cooperation may be investigated. As we have exemplified one may also make a transition from the simpler Prisoner’s Dilemma game by changing a parameter. The model captures the uncertainties on which future situations we may find our opponents and ourselves. The model may easily be extended to include noise in the form of mistakes or misunderstanding, see, e.g., (Molander 1985, Lindgren 1992, 1997). The extension of the model to a spatial setting is also on its way.
References


A Construction of the Total Game FSA

To represent a repeated game between two strategies \( s^a \) and \( s^b \), we construct an FSA \( G \) where each node corresponds to two nodes, one node from each strategy. \( G \) has a node \( G_{ij} \) for each pair of nodes \( (s^a_i, s^b_j) \) in the two strategies, and each edge corresponds to an element in the transition matrix of the game.

The probability of transition between two nodes in \( G \) is a function of the matching rules leading from the first node to the second one, and of the probability distribution of the payoff matrix. This probability distribution is translated to a probability distribution of the joint value of all elementary behaviours for both players (a vector of eight binary values). We have estimated the probability distribution using a Monte Carlo method (we draw samples according to the distribution of the payoff matrix, and count the number of times each joint value of the elementary strategies occur). The joint value representation has the nice property that it is simple to directly compute if the matches or not (if the joint value is represented as an integer, it is a simple matter of shifting and bitwise and operations).

The procedure to compute the transition probabilities consists of executing the following steps for each node in \( G \):

- Generate all edges from the node, and set the transition probabilities of the edges to zero. Every edge in \( G \) corresponds to a pair of edges from \( s^a \) and \( s^b \), including the “match all” in the strategies that there always exists a matching rule in every node.

- Loop over all the joint values, and find the first matching edge (i.e. the first matching edge in \( s^a \) and \( s^b \) separately). As stated above, there is always a matching edge. Then add the probability of the joint value to the transition probability of the edge.

This procedure is repeated for each possibility due to mistakes (four combinations), and the resulting transition probabilities are weighted together according to the mistake probability \( \rho \).

Each node is associated with a pair of elementary strategies, one for the row player and one for the column player. The expected values of all pairs was computed analytically using Mathematica. From this, the expected payoffs with mistakes rate \( \rho \) is:

\[
\text{payoff}(a_i, a_j, \rho) = (1 - \rho)^2 p(a_i, a_j) + \rho (1 - \rho) p(\neg a_i, a_j) + \rho (1 - \rho) p(a_i, \neg a_j) + \rho^2 p(\neg a_i, \neg a_j)
\]  

(7)

where \( p(\cdot, \cdot) \) is the expected payoffs of two elementary strategies, and \( \neg a_i \) denotes the opposite of \( a_i \).