The Advantage of Complexity in Two $2 \times 2$ Games

JIM ENGLE-WARNICK
Department of Economics, McGill University, Montreal, H3A 2T7, Canada

Received July 2, 2003; revised May 2, 2004; accepted May 17, 2004

Competing populations of finite automata co-evolve in an evolutionary algorithm to play two player games. Populations endowed with greater complexity do better against their less complex opponents in a strictly competitive constant sum game. In contrast, complexity determines efficiency levels, but not relative earnings, in a Prisoner's Dilemma game; greater levels of complexity result in mutually higher earnings. With reporting noise, advantages to complexity are lost and efficiency levels are reduced as relatively less complex strategies are selected. © 2004 Wiley Periodicals, Inc. Complexity 9: 71–78, 2004

Key Words: complexity; repeated games; computational experiments; finite automata; genetic algorithm; repeated-game strategies

INTRODUCTION

Theoretical predictions for repeated games suffer from an embarrassment of riches. In interesting modeling situations there are typically many possible equilibria; there may be so many equilibria, in fact, as to render predictions unhelpful when applying them to empirical data. The mechanism generating multiple equilibria is the theoretical consideration of all logically possible strategies when solving the game. If this set of strategies could be reduced sharper predictions would result.

One way to reduce the set of theoretically possible strategies in a repeated game is to assign them a cost for complexity, forcing the agent to trade off complexity for higher payoffs in the game. In economics, finite automata have provided a means to do this. A finite automaton consists of a set of states, a decision plan for each state, and a function that defines transitions between states based on the actions taken by an opponent. The number of states in a finite automaton can serve as a proxy for computational complexity.¹ It has been shown that when agents have preferences over both their monetary payoff and the number of states in the automaton that implements their action plan, the set of possible equilibrium strategies is dramatically reduced.

¹The number of states is by no means the only way to measure complexity, nor is it clear, without empirical evidence, that it is relevant to actual decision-making. See [1] for candidate quantities of complexity, and [2] for an empirical study using data from economics experiments.

Correspondence to: Jim Engle-Warnick, E-mail: jim.engle-warnick@mcgill.ca
[3–5]. It has also been shown that adaptive agents in a single population evolving according to a genetic algorithm alter their long-term behavior when automata are assigned costs for complexity [6].

This article takes a different tack, presenting evolutionary computational experiments that seek to understand the role of complexity in games in a different way. Although it is often taken for granted that as systems evolve through time they become more complex [7], complexity in the situation we study renders the system essentially unsolvable by the agents who must make decisions; with complex strategies, nearly anything is a solution. But if in fact there is no evolutionary advantage to complex strategies, then we could hypothesize that they may be empirically irrelevant and thus refine our searches for the strategies, and therefore the equilibria, that people actually play. Thus we can gain from better understanding the mechanisms that apply selection pressure to complexity in repeated games.

In the numerical experiments, strategies are modeled using finite automata. Two populations of automata endowed with different levels of complexity capabilities compete in a repeated game. The populations evolve across generations of play within a genetic algorithm, which presents each population of strategies to the evolutionary mechanisms of selection, reproduction, and mutation to evolve fitter strategies against the opposing population over time. Using the computational experiments, we are able to observe the effect of differences in complexity among the competing populations.

We study both a Matching Pennies (MP) game (which is a strictly competitive constant sum game) and a version of the well-known Prisoner’s Dilemma (PD). In MP if one agent wins, the other loses; such a game provides a tough test of complexity because it is difficult to obtain an advantage over an opponent in it. We study the PD to determine whether the results are robust to a game that models a social situation, to test a game with a starkly different set of theoretical solutions, and to compare the results to the plethora of existing knowledge of play in this game. Both finite automata and genetic algorithms have been used to study play in repeated games (e.g. [7, 9]).

In MP we find that there is indeed an advantage to complexity: populations with greater complexity capability do better than their opponents. We find that this advantage diminishes as the environment becomes noisy, because evolution selects less complex automata to avoid over-fitting noisy data. In the PD we find that the level of complexity available in the populations is positively related to the efficiency of the outcomes, but that relative outcomes are not affected. Although complexity is not a strategic advantage, its presence, surprisingly, improves outcomes for both populations. Again, reporting noise diminishes the effect because less complex strategies are selected in its presence.

The next section of the article describes the repeated games, the strategy model, and the simulations. The following section describes the results. The final section concludes.

THE SIMULATION ENVIRONMENT

The Repeated Games

We study two games: Matching Pennies (MP) and a Prisoner’s Dilemma (PD). The MP game, shown in Figure 1A, has the property that whenever one player wins the other loses. The rows in the figure designate the choices available for one player (up and down) and the columns for the other player (left and right). The numbers in the cells represent the payoffs for the row and column player, respectively. For example, if the players choose the actions up and left, then the row player receives a payoff of 20 and the column player receives 0. The equilibrium of this game is for both players to play each of their strategies with probability one-half.

The PD, shown in Figure 1B, is identical to the one studied by Miller [9]. This game has the well-known property that if the opponent cooperates, then a player is better off defecting, which results in a unique equilibrium with both players defecting in the game when it is played once. By the Folk Theorem of Repeated Games [10], it can be shown that any payoff that results in an average per round payoff of at least 10 for each player is sustainable in equilibrium, if the players are risk neutral and sufficiently patient and the game is played infinitely long. This result is relevant here because finite automata cycle when they play against each other, and the time horizon in the numerical experiments will be much longer than the cycles.

Both of these games have been studied in the experimental laboratory, where human subjects play the game repeatedly for pay. The stylized fact for games analogous to

![FIGURE 1](image)

(A) The Matching Pennies game. (B) The Prisoner’s Dilemma game.
MP is that the marginal frequency of play is consistent with equilibrium predictions, but actions are serially correlated rather than random as the theory predicts (e.g., [11–13]), suggesting some kind of nonrandom strategic behavior underlying the actions. In PD games with repeated play, cooperative norms have been seen to emerge (e.g., [14–17]). However, little is known regarding the strategies that people are using when they play the games, and this article is a step toward developing a theory of strategy selection.

The Strategies
We model strategies with finite automata (called Moore machines), which consist of a finite set of states, a decision plan for each state, and a transition function that determines the next state after observing an opponent’s decision. Finite automata have been used to model strategies in the theory of repeated games [5], have been used to describe human behavior in economics experiments [2], and can model behaviors deemed important in repeated games including punishing an opponent for defection and counting until the game nears its end.

The finite automata in this study are computationally represented as objects that contain tables of integers. Each object contains three tables, and each table represents the same automaton’s plan of action in three different ways. Figure 2A presents a randomly generated automaton for the PD. The rows represent the states; this automaton has four rows, thus four states. One of the rows, Row 3, is designated the starting state. The columns underneath the heading “Next State” specify the next state conditional on the current state and the opponent’s action. The column on the far right labeled “Current State Action” shows the action to take in the current state; C represents the action cooperate and D represents defect in the PD. So, for example, this automaton begins the game in State 3 playing C. If the opponent plays its action C, the automaton transitions back to the same state where it plays C again. If the opponent plays its action D, this automaton transitions to State 2 where it responds with action D. Notice that regardless of the opponent’s actions, this automaton will always find itself either in State 1 playing C or in States 2 or 3 playing D.

A brief look at Figure 2B reveals that States 1 and 2 are redundant (i.e., equivalent). The automaton would behave the same if these two states, from which the automaton cannot escape, and which play the same action, were merged into one. Figure 2C presents the third table contained by the automaton object: the minimal state representation of the automaton. This turns out to be the well-known grim-trigger strategy that cooperates as long as an opponent cooperates and punishes forever if the opponent ever defects.

The Evolution of Strategies
We randomly generate two separate populations of finite automata. Each population consists of 30 finite automata, and each automaton is created by randomly inserting integers into its random table (as shown in Figure 2A), and randomly selecting a starting state. Let $M$ and $N$ denote the two populations, and let $M_i$ represent an automaton in population $M$ and $N_j$ an automaton in population $N$. Each

\[\text{FIGURE 2} \]

(A) Randomly drawn finite automaton state transition and action table.
(B) Connected finite automaton state transition and action table.
(C) Minimal state finite automaton state transition and action table.

A finite automata theory provides a precise method to construct minimal state machines, see [19, 20] for the theory. In this article the minimal state representation is used for analysis of behavior, while the random state representation is operated upon by the genetic algorithm.

---

3The simulations were programmed in Ox using object-oriented programming [18].
\[ M, i = (1, 2, 3, \ldots, 30), \] plays a repeated game against each \[ N, j = (1, 2, 3, \ldots, 30). \] Thus although populations cannot differ in roles (because the games are symmetric), one can think of one population \( M \) playing the row position and population \( N \) playing the column position.

In these tournaments the sum of the received payoffs is computed as a basis for an automaton’s evolutionary fitness. Upon completion of the tournament the two populations are independently subjected to the evolutionary operations of reproduction, breeding, and mutation, where the probability of selection increases as the automaton’s fitness is increasingly better than the average fitness of its population. The algorithm we use is well known, and a brief description follows.\(^5\)

The raw fitness \( y_i \) of an automaton is the sum of its payoffs in each round of every game it plays. The raw fitness is normalized by the formula \( \bar{y}_i = (y_i - \bar{a})/d + a \), where \( y_i \) is the sum of the payoffs in each period of the tournament for automaton \( i \), \( \bar{a} \) is the sample mean, \( d \) is the standard deviation, and \( a \) (set equal to 2) is a scaling parameter that determines the importance of the relative performance of the automaton. This formula, which determines the probability that an automaton will be selected for a genetic operation, guarantees that a machine that performs worse than two standard deviations from the mean is not selected. Thus automata are selected according to their fitness compared to the mean population fitness, and automata more than two standard deviations below the mean are never selected.

An automaton selected for reproduction is copied exactly as is into the next generation. After selection, pairs of automata are repeatedly selected to breed two children. A cross-over element in the random table is randomly selected, dividing the table into two parts, and the elements in the table of one parent beginning with the upper left element, counting left to right across the columns and top to bottom down the rows up to the cross-over element are combined with elements after the cross-over point from the table of the other parent to create a child. From the new random tables, the connected and minimal state tables are then computed. The minimal state table is used to examine the effective complexity that exists in the population. The children are then subjected to mutation, with each integer in the table altered with a small probability by randomly drawing a valid integer to replace the integer in the table.

The new population consists of the set of replicates and the set of children produced through crossover and mutation. The tournament is repeated in the new population, after which selection occurs as before. Selection pressure applies to the populations to produce better fitting automata against the competing population across generations.

Each population consisted of 30 automata. In each generation of the simulation 20 automata were selected for replication, and 10 for breeding. Maximal complexity was 8 states (results are robust up to 16 states but do not provide any additional intuition), and the automata played 80 round games in the tournaments. Because automata enter into cycles when playing against each other and the cycles cannot be longer than the number of states of the smaller of the 2, 80 rounds were sufficient to capture long-term behavior. Convergence was achieved well before 50 generations of evolution, which is where the simulations were terminated; the results in the 50th generation are presented below. The average from 50 simulations is presented.

The experimental design is such that a population of one state machines plays against populations of maximum size one through maximum size eight states. A population of maximum size two states plays against populations of maximum size two through eight. This continues up through populations of maximum size eight machines playing against each other.\(^6\)

There were two experimental treatments. In the first treatment opponent actions were reported accurately in every round. In the second treatment, with a small probability the opponent’s action was incorrectly reported.\(^7\) This treatment tests the robustness of the results to reporting noise.

**RESULTS**

**Matching Pennies**

Figure 3 presents the results for MP. The graphs present average per round payoffs for populations of maximal complexity one, two, three, and eight (labeled by the legend; complexities four through seven are deleted for clarity in

\(^5\)This algorithm was used by [9], and is used here to assure that the results are comparable with existing established results (the tabular chromosome representation and algorithm qualitatively replicate the results in [9]). Results here are robust to a variety of fitness, crossover, and mutation schemes as well as representations of the chromosomes (in particular, a binary representation). For newer genetic algorithm technology see, e.g. [21].

\(^6\)The number of simulations was largely determined by computational time: each simulation embedded \( 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 36 \) simulations.

\(^7\)The probability of an incorrect report was 5% for the constant sum game and 1% for the Prisoner’s Dilemma game. The Prisoner’s Dilemma was more sensitive to noise and performance degraded to such an extent at 5% noise as to not be able to interpret its marginal effect on play.
Figures 3 and 4) against populations of equal and greater complexity (labeled by the horizontal axis).

For example, a population with a fixed maximal size of one state played against populations of maximal size one through eight states; the average per round payoff this population received is shown by the bottom dashed line in Figure 3A labeled "1 state" by the key, which begins near an average payoff of 10 when both populations have a maximum size of one and then drops to around 2 per round when the opposing population contains two-state machines. The figure shows that the results do not change as the number of states in the opposing population increases from two through eight. The middle dashed line, labeled "2 states" by the key, shows analogous results when a population of fixed maximum size two states plays against populations of maximum size two through eight states.

Result 1

There is an advantage to complexity in repeated matching pennies games.

Figure 3A reveals that Result 1 is most stark when a population has a maximum size of one state. When this population plays the game against a population containing machines with more than one state, the average payoff is about 2 per round (in this constant sum game, the opposing population’s payoff must be $20 - 2 = 18$ per round). This is because it takes no more than a two-state machine to learn how to best respond to a one-state machine; additional states are not necessary to improve relative performance. The more complex population cannot totally exploit the less complex population because selection is probabilistic, permitting some machines to be reproduced that are not fit against the less complex population and because the reproduction and mutation also produces some machines that are not useful. Nevertheless, the more complex population takes home an average pay of 18 versus an average pay of 2 for the one-state population.

The magnitude, but not the qualitative aspect of this advantage decreases as the less complex population increases in complexity: the lines in Figure 3A move toward the northeast corner of the graph, indicating more even outcomes as the disadvantaged population gains in complexity. And holding the weaker population’s complexity constant, average earnings decrease as the competing population complexity increases: the lines are downward sloping from left to right. For example, the lowest average earnings of the two-state population occurs against the eight-state population (7.35 per round, implying 12.65 per round for the opposing population), and the lowest average earnings of the three-state population also occurs against the eight-state population (9.22 per round, implying 10.78 per round for the opposing population). Statistical significance is lost when the maximum four-state population plays against populations of size four through eight.

© 2004 Wiley Periodicals, Inc.

A mechanism was identified in [22] that was responsible for the advantage to complexity in an adaptive environment: with equal complexities adaptive agents essentially take turns from generation to generation with the payoff advantage in a strictly competitive game, cycling back and forth as each best-responds to what worked well in the previous generation. With an extra state of complexity, an agent can sometimes trap her opponent and reverse the normal reversal of fortune that occurs between the populations from generation to generation.
Result 2
Reporting noise diminishes advantage to complexity in repeated MP games.

With reporting noise, the advantage to complexity decreases; this can be seen in Figure 3B and 3C. Figure 3B shows that qualitatively the advantage to complexity remains, but across the board the magnitude of the advantage decreases; the visible lines in the graph shift upward, indicating increasing earnings for the disadvantaged population and thus decreasing earnings for the advantaged. Figure 3C presents the difference between the 5% reporting noise treatment and the no noise treatment; in every case in which the disadvantaged population has one through four states, this population earned slightly more when the reporting noise was added.

The results indicate that complexity is an advantage in the constant sum game, and this advantage decreases with reporting noise.

Prisoner’s Dilemma Game
Figure 4 presents the results for the PD game and includes payoffs for both populations because, unlike in the constant sum game, one cannot infer the payoff of one population from that of the other. The panels on the left show the payoffs for the population with fixed maximum complexity, exactly as before in Figure 3, and the right column presents
the payoffs of the opposing population. The figures are interpreted in the context of the appropriate population; thus in the left-hand panels, the horizontal axis represents the opposing population complexity and the legend the population’s complexity, whereas in the right-hand panels, the horizontal axis represents the population’s complexity and the legend indicates the opposing population’s complexity. All other elements are presented as before.

Result 3

Complexity does not result in a relative increase in payoffs, but increasing complexity of either population increases the efficiency of the outcome.

This surprising result can be gleaned by noting that per-round payoffs in both the left and right panels of Figure 4 are nearly identical and that payoffs increase as the complexity of the weaker population increases from one state to more than one state. For example, the average payoff for both populations is just over 10 per round when the population in the left-hand panels is limited to one-state machines. The payoff jumps to approximately 15 when two-state machine populations play against each other, and, holding one population constant at two states, payoffs actually increase to about 25 as the opposing population maximal complexity is increased to eight. Not only that, but payoffs increase when the advantaged population increases in complexity: the lines in the graph slope upward to the northeast. In contrast to MP, a higher relative complexity does not translate to a relative advantage in this game, but it does increase efficiency.

Result 4

Reporting noise reduces the payoffs of both populations.

With reporting noise, as with the constant sum game, payoffs drop across the board when the population in the left-hand panels contains more than one state. Now it takes three states to achieve a significant improvement in payoffs over the case in which the population in the left-hand panels is limited to one state (without noise improvement only required the addition of the second state).

Figure 4C shows the difference between the case with reporting noise and without and appears to be noisy, which is the case because above one state there is very little quantitative or qualitative difference in payoffs when the disadvantaged population has more than one state. The difference in both cases is significantly less than zero, indicating that reporting noise had a negative effect on outcomes.

Results from the PD game indicate that the more complexity exists in the environment, the better the outcome for both populations. This type of advantage to complexity is diminished with reporting noise.

DISCUSSION

The role of reporting noise has been established in the evolution of finite automata in single populations [9]: less complex strategies are selected to avoid over-fitting noisy data. However, these two-population experiments reveal a differential effect of noise with regard to the evolution of complexity in the two environments. Figure 5 presents the average complexity of the minimal state representations of the evolved machines in the advantaged populations against populations with maximum complexity of one to eight for both games. There is no difference in complexity in MP whether noise is present or not; however, machines are less complex in the face of noise in the PD when the disadvantaged population has a maximum of three states or less.

We speculate that the role of complexity in the PD is related to the number of useful simple punishment strategies available for selection. Such strategies are useful in the PD, where one can hold down the other player’s payoff, but not in MP, where one cannot. Both Miller [9] and Arthur [7] report that the evolution of punishment strategies is correlated with the emergence of cooperation in the PD. When one population is limited to one state, there are no punishment strategies available to it and the system evolves to mutual defection. Adding a state makes, among other strategies, the Grim Trigger strategy (illustrated in Figure 2) available to punish defections, and this type of strategy is selected increasingly over time. A surprising result here is that even when the opposing population’s complexity increases, holding the less complex population constant, efficiency improves even more (the lines in Figure 4 are upward sloping). We speculate (and simulation results tend to indicate, though a suitable metric has proven elusive) that
the availability of more punishment strategies for selection results in a mutual advantage for both populations.9

CONCLUSION
In this article we explored the advantage of complexity in a strictly competitive game and in a game representing a social dilemma. The innovation in this article is the exploration of two separate populations of finite automata playing repeated games, making it possible to explore the effects of heterogeneity between the populations.

In the strictly competitive environment of MP, complexity can be an advantage when agents are modeled as finite automata learning according to a genetic algorithm. This advantage diminishes with reporting noise. In the PD, complexity plays a different role, where efficiency in outcomes is essentially controlled by the less complex population. This feature, combined with the fact that efficiency improved holding the disadvantaged population complexity constant while increasing the complexity of the opposing population, suggests a different role for complexity in social dilemmas than in strictly competitive environments. Evolution selects punishment strategies that in turn sustain cooperation, and it appears that the more punishment strategies available, the better the outcomes.

The goal of the study was to find a theoretical reason to reduce the set of all logically possible strategies for the purpose of determining the empirical relevance of strategies in repeated games; these results suggest that complexity may indeed be useful, providing the environment is sufficiently deterministic. The results suggest that if people do not consider a vast number of strategies when making decisions, then the existence of noise in the environment may justify a reduction of the strategy set.

9It could also be the case that populations converge more quickly when the difference in complexities increases, which would leave open the possibility that in the long term, this effect would be lost. Spot checks of several cases run for 250 generations did not overturn this result. Presenting the results after 50 generations is consistent with existing literature.

REFERENCES