Beating A Finite Automaton in the Big Match

Lance Fortnow*
Peter Kimmel†

Department of Computer Science
The University of Chicago
1100 East 58th Street
Chicago, Illinois 60637

Abstract

We look at the Big Match game, a variation of the repeated Matching Pennies game where if the first player plays tails the game ends with the first player receiving the last round's payoff. We study this game when the second player is implemented as a finite automaton. We show several results including:

- If the first player knows the number of states of the second player's automaton then he can achieve the maximum score with a deterministic polynomial-time algorithm.
- If a deterministic first player does not know the number of states of the second player then he can not guarantee himself more than the minimum score.
- If we allow player one to run in probabilistic polynomial-time then he still cannot achieve the maximum score but he can get arbitrarily close.
- In a slight variation of the Big Match, the first player cannot have an even close to dominant strategy.

1 Introduction

The king has a dilemma. Every year the crowned prince presents him with a sack of gold or a sack of stones. The king always loves to receive the gold but sees the stones as a slap in the face. Before the prince makes the presentation, the king has the option to order the guards to kill the prince. If the guards kill the prince, they will then tear open the sack to reveal the contents. If the sack contains stones then the king has justifiably eliminated the pesky prince. If, however, the sack contained gold then the masses will force the king from his throne for killing a prince bearing wonderful gifts.

*Email: fortnow@cs.uchicago.edu. URL: http://www.cs.uchicago.edu/~fortnow.

Partially supported by NSF grant CCR 92-53582.

†Email: kimmel@cs.uchicago.edu.

L.S. Shapley [S] introduced the notion of stochastic games in 1953 as a generalization of repeated games. He showed that finite discounted zero-sum games have a value, and that both players have stationary optimal strategies. In 1957, Gillette [G] considered a different kind of stochastic games called undiscounted stochastic games. He introduced the game above, The Big Match (though not called that until [BF]), to show that undiscounted stochastic games need not have a solution with both players having stationary strategies.

After Gillette, much work was done to find sufficient conditions for the existence of a value in undiscounted stochastic games. This question remained open until it was shown by J.F. Mertens and A. Neyman [MN] in 1981 and independently by Monash [M] in 1979 that all undiscounted stochastic games do in fact have a value.

Although these games have a value, they need not have optimal strategies. In 1968, Blackwell and Ferguson [BF] showed that for the Big Match, the king has no optimal strategy, and that in order for the king to achieve a close to optimal final payoff, he must use a strategy that depends on the entire history of the game.

We are interested in continuing the work of Fortnow and Whang, by extending it to more general stochastic games. In this paper, we look at the "bounded rationality" version of the Big Match where the prince has only a small memory. We model the prince by a deterministic finite automaton and the king by a polynomial-time computer. We view the Big Match as a variation on the Matching Pennies game studied by Fortnow and Whang.

We show that if the king (player P1) knows the number of states of the automaton that models the prince (player P2), then the king can guarantee himself either always receiving gold or justifiably killing the prince. If a deterministic king does not know the number of states of the prince, then the king should learn to enjoy stones.

If we allow the king some randomness then the situation changes dramatically. Even if the king has no idea of the size of the prince's memory, the king can still achieve near optimal conditions.

We also show that in a small variation of the big match, even a randomized king cannot achieve a close to optimal situation.

2 Preliminaries

Consider the two player game *Matching Pennies* played by players P1 and P2: In each round P2 chooses an element of {Heads,Tails} and P1 tries to predict P2's choice, winning 1 point if he is correct, 0 points if he is not. P1 tries to maximize his score, while P2 tries to minimize P1's score. We describe each round of this game by the following *payoff matrix*, in which P1's moves are represented by lower case letters, P2's moves by upper case letters, and the entries are the payoffs to P1:

Matching Pennies:

	H	T
h	1	0
t	0	1

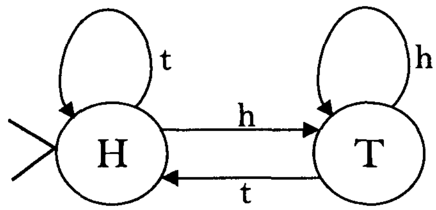

Figure 1: An automaton that plays the "change when caught" strategy.

We are interested in the case where the game is played for an infinite number of rounds. In this case, we define the *final payoff* for P1 to be

$$\limsup_{n\to\infty} (a_1 + \cdots + a_n)/n, \tag{1}$$

where a_i is the payoff received in the ith round. If the players are allowed to use randomization, then the final payoff is the expected value of Expression (1).

A *strategy* σ_2 for P2 is a function $\sigma_2(\{h,t\} \times \{H,T\})^* \to \{H,T\}$. An element of the domain is a *history* of the game played so far; σ_2 maps these histories to next moves. A strategy for P1 is defined analogously. A *recursive strategy* is a strategy computable by a Turing machine. A *polynomial-time strategy* is a strategy computed by a Turing machine which computes the next move in polynomial-time in the length of the history.

In this paper we will be mainly concerned with regular strategies. A *regular strategy* is a strategy realized (or played) by a deterministic finite automaton (DFA). We say a deterministic finite automaton M *plays* Matching Pennies (for P2) if each state of M is labeled with an H or a T (the move to be played), and each state has exactly two transitions from it, one labeled with h, the other with t (the move played by P1).

For example, Figure 1 shows an automaton that realizes the "change when caught" strategy for P2, which is to start out by playing H and then switching moves whenever P1 wins a point.

Let S_1 be a set of strategies for P1, and let S_2 be a set of strategies for P2. Let $\sigma_1 \in S_1$, and $\sigma_2 \in S_2$. Let $\pi(\sigma_1, \sigma_2)$ denote the final payoff to P1 given that P1 and P2 play the strategies σ_1 and σ_2 respectively. We say that a strategy $\sigma_1 \in S_1$ is

1. *optimal for S_1 against σ_2* if for every $\sigma_1' \in S_1$, $\pi(\sigma_1, \sigma_2) \geq \pi(\sigma_1', \sigma_2)$.
2. *ϵ-optimal for S_1 against σ_2* if for every $\sigma_1' \in S_1$, $\pi(\sigma_1, \sigma_2) \geq \pi(\sigma_1', \sigma_2) - \epsilon$.
3. *dominant for S_1 against S_2* if for every $\sigma_2' \in S_2$, σ_1 is optimal for S_1 against σ_2'.
4. *ϵ-dominant for S_1 against S_2* if for every $\sigma_2' \in S_2$, σ_1 is ϵ-optimal for S_1 against σ_2'.

Note that if the phrase "for S_1" is left out of the above definitions, then S_1 is assumed to be the set of all strategies. Similarly, if the phrase "against S_2" is left out, then S_2 is assumed to be the set of all strategies.

Fortnow and Whang [FW] show the following results about the Matching Pennies game:

Theorem 2.1 *There is a recursive strategy σ_1 which is dominant (for the set of all strategies) against the set of polynomial time strategies.* □

Theorem 2.2 *There exists a polynomial-time strategy that is dominant against the set of regular strategies and converges in a polynomial number of rounds.* □

Theorem 2.3 *There exists a behavioral regular strategy σ_2 for which P1 has no optimal strategy, but $\forall\epsilon > 0, \exists$ ϵ-optimal strategy for P1.* □

In Theorem 2.2, the converging refers to how fast the lim sup in the final payoff converges to optimal, and in Theorem 2.3, a *behavioral regular strategy* is a strategy realized by a finite automaton which is allowed to flip coins and move to new states based on this randomization.

The game we consider in this paper is a slight modification of Matching Pennies, called the *Big Match*. The Big Match works the same way as Matching Pennies except for the following catch: If P1 ever plays t, the game is over, with the final payoff to P1 being 1 if P2 played T in the last round and 0 if P2 played H. If P1 always plays h, the game continues forever, and the final payoff to P1 is calculated as in the Matching Pennies game. In other words, the goal for P1 is to predict correctly a round in which P2 will play T.

The Big Match game can be described by the following payoff matrix:

The Big Match:

	H	T
h	1	0
t	$0\star$	$1\star$

If P1 ever receives a starred payoff, then the game is over and that is his final payoff.

The games examined by Fortnow and Whang are repeated single matrix games. They are a special case of stochastic games, which are repeated games played on any number of payoff matrices: after a round is played with one payoff matrix, the players play the next round at another (or possibly the same) payoff matrix, which is determined by a probability distribution which is in turn determined by the previous matrix and players' moves. The Big Match is also a stochastic game, which involves three simple payoff matrices.

More formally, A *finite stochastic game* (as defined by Blackwell and Ferguson) consists of three non-empty finite sets S, I, J, a real valued function a defined for all triples $(s, i, j) \in S \times I \times J$, and a function p which associates with each triple (s, i, j) a probability distribution $p(\cdot|s, i, j)$ on S. An *initial state* $s_0 \in S$ is known to both P1 and P2. P1 chooses an $i \in I$, and simultaneously, P2 chooses a $j \in J$. P1 is awarded $a(s, i, j)$ points and the game moves to state s' selected according to $p(\cdot|s, i, j)$. The new state s' is announced to both players, who again choose elements of I and J and again receive payoffs. They then move to another state and the process continues. The *final payoff* to P1 is defined by

$$\limsup_{n\to\infty} \,(a_1 + \cdots + a_n)/n, \tag{2}$$

where a_i is the payoff P1 receives in the ith round. Again, if the players are allowed to use randomization, then the final payoff is the expected value of Expression (2).

Remark: When Shapley introduced the notion of stochastic games in 1953, he considered a different final payoff function than the one above, called the *β-discounted payoff*. For $\beta \in \{0,1\}$, the final discounted payoff to P1 is $\sum_{i=0}^{\infty} \beta^i a_i$, where a_i is the payoff to P1 in round i. Shapley showed that finite discounted zero-sum games have a value, and that both players have stationary optimal strategies. (In a zero-sum game, there are reward functions for both players, r_1, r_2 that have the property that $r_1 = -r_2$. Since we consider optimal strategies for P1 only, we have disregarded the payoff function of P2. See also [S] and [PTV].) Gillette [G] in 1957 was the first to consider undiscounted stochastic games, i.e., the stochastic games in which the final payoff function is the average of the reward per round, as we described above.

We can see that Matching Pennies is a finite stochastic game with $|S| = 1$. We can also redefine the Big Match as the finite stochastic game with

$$\begin{aligned} S &= \{0,1,2\}, & I = J &= \{0,1\}, \\ a(2,i,j) &= \delta_{ij}, & a(s,i,j) &= s \text{ for } s = 0,1, \\ p(2,0,j) &= 2, & p(2,1,j) &= j, \\ p(s,i,j) &= s \text{ for } s = 0,1, & s_0 &= 2, \end{aligned}$$

where $p(s,i,j) = t$ means that given (s,i,j), the next state is t with probability 1.

Blackwell and Ferguson [BF] show that for all $\epsilon > 0$, P1 can achieve an expected final payoff of at least $\frac{1}{2} - \epsilon$ in the Big Match with a polynomial-time strategy, but for no strategy is P1 guaranteed an expected final payoff of exactly $\frac{1}{2}$. This means that there is no dominant strategy for P1, but for all $\epsilon > 0$, P1 has a polynomial-time ϵ-dominant strategy. In the next section, we will see how P1 can improve his expected payoff if P2 is played by a DFA.

3 Results

We consider the Big Match for the case where P2 is restricted to (deterministic) regular strategies. In this case, we will often refer to P2 as a DFA. The first three theorems examine the effect of (1) knowing the number of states P2 has and (2) using randomization on P1's ability to dominate in the Big Match. The last theorem considers the effect of a slight modification of the payoffs of the Big Match on P1's ability to dominate.

Theorem 3.1 *For every k, there exists a deterministic polynomial-time strategy σ_1 for P1 that achieves the maximum final payoff of 1 against any k-state strategy for P2.*

Proof: Let P2 be a DFA with k states. P1 plays h for $3k$ rounds. Since P2 has only k states, P2 will enter a cycle that will be repeated as long P1 plays h. Furthermore, P1 can determine what this cycle is by examining the first $3k$ moves of P2.

Suppose this cycle includes a T. Then P1 continues to play h through another iteration of the cycle up until the point where he knows T is the next move in P2's cycle. Then P1 plays a t to match P2's T and achieves a final payoff of 1.

In the case where this cycle includes no T's, P1 simply plays h forever. This means that P2 will never exit his cycle, which consists solely of H's. Thus P1 receives a payoff of 1 each round, and hence a final payoff of 1. □

Theorem 3.1 says that there is a polynomial time strategy σ_1 which will be optimal against any DFA M as long as σ_1 is given the number of states of M as an additional input. We can say that with some help, σ_1 dominates the set of regular strategies. What happens if P1 is not told the number of states of P2?

Theorem 3.2 *Let P2 be a DFA M and suppose P1 knows nothing about M except that it is a DFA. If P1 is deterministic, then he cannot guarantee himself a positive final payoff.*

Proof: Assume P1 can guarantee a positive final payoff. Suppose P1 plays h forever. If M is a DFA which after some round always plays T, then the final payoff to P1 will be 0. Therefore, in order to guarantee a positive final payoff, P1 must play t at some point.

When P1 plays t, the game will end and P1's final payoff will be 0 or 1 depending on whether P2 plays T on the last round. Thus our assumption implies that P1 can predict a T for P2 at some point in the game. Let r be the round at which P1 plays t. It is possible that M has an initial path of $r + 1$ states before any cycling occurs. But this implies that P1 knows the label of a state of M that has not yet been entered, which contradicts the assumption that P1 knows nothing about M. □

The situation changes dramatically if we allow P1 to use randomization.

Theorem 3.3 *For all $\epsilon > 0$, there exists a probabilistic strategy σ_1 for P1 that achieves an expected final payoff of $\geq 1 - \epsilon$ against any DFA for P2. Furthermore, σ_1 computes its next move in time polynomial in both the size of the history and the number of states of P2. If P1 eventually plays t it will do so in expected number of rounds polynomial in the number of states of P2 and $1/\epsilon$.*

Proof: Let $\epsilon > 0$. Let σ_1 be defined by the randomized algorithm given in Figure 2.

Let us say that P1 is *tricked* if the game ends in Case 2.2. Note that if P1 is tricked then P1's final payoff is 0.

Lemma 3.4 *If P1 is never tricked, he achieves a payoff of 1.*

Proof: Say P1 is never tricked. One of three outcomes must occur during the ith iteration:

1. P1 repeats steps 20 and 30 forever, never reaching step 40. This means P1 always reaches the ELSE of step 30 and goes back to step 20 with a bigger value for k. Eventually, k will be bigger than the number of states of P2. At this point, while P1 plays h for $4k$ moves in step 20, P2 will be stuck in a cycle. Since P1 always reaches the ELSE of step 30, this means that P2 is in a cycle which does not contain a T. Therefore, P1 will always go back to step 20 and play h forever, and P2 will remain stuck in its all-H cycle, so P1's final payoff will be 1.

2. P1 repeats step 40 forever. Once per cycle, P1 plays t with probability $p = \epsilon/2^i$. Since this occurs infinitely often, with probability 1, P1 will play t at some point. At this point, P2 will also play T, because by assumption, P1 is not tricked. Therefore P1 will receive a final payoff of 1.

ALGORITHM σ_1:

10: LET $k := 10, p := \epsilon/2, i := 1$.
    (k estimates the number of states of P2, i is the iteration, p is a probability.)
20: PLAY h for $4k$ rounds.
30: IF P2 appears to be in a cycle at the end of the $4k$ rounds
    (i.e., the pattern of P2's moves starts repeating for at least k rounds)
    AND this cycle contains a T,
  THEN GOTO 40.
  ELSE, LET $k := 10k$ and GOTO 20.
40: PLAY h UNTIL either
    CASE 1. P2 deviates from the detected cycle. In this case, GOTO 50.
    CASE 2. The next move in the detected cycle is T. In this case, PLAY t with probability p.
      CASE 2.1. P1 plays t, P2 plays T.
        Game ends with final payoff 1 to P1.
      CASE 2.2. P1 plays t, P2 plays H. (P2 deviates from detected cycle).
        Game ends with final payoff 0 to P1.
      CASE 2.3. P1 plays h, P2 plays T. GOTO 40.
      CASE 2.4. P1 plays h, P2 plays H. (P2 deviates from detected cycle.) GOTO 50.
50: LET $k := 10k, p := p/2, i := i + 1$.
60: GOTO 20.

Figure 2: Probabilistic polynomial-time strategy which is ϵ-optimal against regular strategies in the Big Match.

3. P1 reaches step 50, updates k to be $10k$, and moves to iteration $i+1$. In this case, one of these three outcomes must occur in iteration $i+1$.

The important thing to note is that eventually, one of the first two outcomes must occur. Suppose outcome (3) occurred forever. Then eventually k would be bigger than the number of states of P2, and the detected cycle would be a real cycle. P2 could not deviate from this cycle, and thus we could not reach step 50 again, as deviation from the detected cycle is the only way to reach step 50.

Therefore, there exists a i for which the ith iteration of the algorithm results in outcome (1) or (2). Therefore, if P1 is never tricked, then P1 achieves a final payoff of 1. □ Lemma 3.4

During each iteration, P2 has only one opportunity to trick P1: once P2 plays H when P1 suspects a T should be played, P1 moves to the next iteration. Therefore, the probability of P1 being tricked during iteration i is simply the probability that P1 plays t for the round where P2 is attempting the trick, which is simply $\epsilon/2^i$. So the probability that P1 is ever tricked is $\leq \sum_{i=1}^{\infty} \epsilon/2^i = \epsilon$, and thus the probability that P1 is never tricked is $\geq 1 - \epsilon$.[1]

[1]This is not an equality because P2 might not try to trick P1 after a certain iteration.

Therefore, by the above Lemma, P1 achieves an expected final payoff of

$$1 \cdot \Pr(\text{P1 is never tricked}) + 0 \cdot \Pr(\text{P1 is tricked}) \geq 1 \cdot (1 - \epsilon) = 1 - \epsilon.$$

Now consider the running time of this algorithm. The only time that the algorithm needs more than constant time to compute its next move is for the moves that occur immediately after step 20. Here we must examine the history of the game for cycles. Clearly this can be done in time polynomial in the length of the history. However, also note that we need only examine the last $4k$ moves for the cycle. The proof of Lemma 3.4 shows that the value of k will never exceed 100 times the number of states of P2. Therefore, any move can also be computed in time polynomial in the number of states of P2.

Now suppose P1 plays t at some point which must happen in CASE 2. Since the cycle has length k, P1 will play t with in expected $k/p = k2^i/\epsilon$ rounds. Since $k \geq 10^i$ the expected number of rounds in polynomial in the number of states of P2 and $1/\epsilon$. □ Theorem 3.3

Corollary 3.5 *If S_2 is any countable computable enumeration of recursive strategies for P2, then for all $\epsilon > 0$, there exists a recursive, probabilistic strategy σ_1 for P1 that achieves an expected final payoff of $\geq 1 - \epsilon$ against any strategy in S_2.*

Proof: We use the same style algorithm as before. Let $\sigma_2(i)$ be the ith strategy in some enumeration of S_2. Assume P2 is using $\sigma_2(1)$ until we see a deviation from it. Then find the next i such that $\sigma_2(i)$ is consistent with what P2 has played so far. Always play heads until the strategy we are currently considering says P2 will play T next. Then play t with probability $\epsilon/2^{j+1}$, where j is the number of times P2 has caused P1 to consider a different strategy.

Note that in this strategy cannot be computed in polynomial time, but is computed by a Turing machine that has access to another machine that enumerates S_2. □

Theorem 3.3 shows that for any $\epsilon > 0$, there is a probabilistic polynomial time strategy σ_1 for P1 that ϵ-dominates the set of regular strategies. However, we shall see that no strategy for P1 is dominant for the class of probabilistic polynomial-time strategies against the set of regular strategies.

Theorem 3.6 *For every strategy σ_1 for P1, there exists an $\epsilon > 0$ and a DFA σ_2 for P2 such that $\pi(\sigma_1, \sigma_2) \leq 1 - \epsilon$.*

Note: The following proof is based on a similar result in Blackwell and Ferguson [BF], who in turn credit the argument to Lester Dubins.

Proof: Suppose P2 always plays T. If P1 never plays t with positive probability, then P1 receives a final payoff of 0. Therefore, let us assume that at some point, P1 plays t with positive probability. Let $m \geq 0$ be the smallest initial number of T's P1 sees P2 play after which P1 plays t with positive probability $\epsilon > 0$. Suppose P2 plays $T^m H^*$ (m T's followed by H's thereafter). Then P1's final payoff will be 0 with probability $\geq \epsilon$, and therefore the expected value of P1's final payoff at most $1 - \epsilon$. □

Let U be the set of probabilistic polynomial-time strategies for P1.

Corollary 3.7 *No strategy for P1 is dominant for U against the set of regular strategies.*

Proof: Assume σ_1 is dominant for U regular strategies. This implies that σ_1 is optimal for U against every DFA for P2. Theorem 3.6 says there is an $\epsilon > 0$ and a DFA σ_2 for P2 such that $\pi(\sigma_1, \sigma_2) \leq 1 - \epsilon$. However, Theorem 3.3 says there exists a strategy σ_1' in U that achieves a payoff of at least $1 - \epsilon/2$. Therefore, σ_1 is not optimal for U against σ_2, and hence σ_1 is not dominant for U against the set of regular strategies. □

Finally, we give a variation of the Big Match game for which there is an $\epsilon > 0$ for which P1 cannot even ϵ-dominate the class of regular strategies. We define the game *Big Match II* by the following payoff matrix:

The Big Match II:

	H	T
h	2	0
t	$0\star$	$1\star$

The only change is that P1 receives a payoff of 2 instead of 1 if both players play heads. As before, if P1 ever receives a starred payoff, then the game is over and that is his final payoff.

Theorem 3.8 *There is no (1/2)-dominant strategy for P1 against the class of regular strategies for P2 in the Big Match II game.*

Proof: Assume not. Then there exists a strategy σ_1 for P1 such that for all strategies σ_1' for P1 and all regular strategies σ_2 for P2, we have

$$\pi(\sigma_1, \sigma_2) \geq \pi(\sigma_1', \sigma_2) - 1/2. \tag{3}$$

Consider P1's strategy against the strategy T^* for P2 (P2 plays T forever). If P1 always plays h, then his final payoff will be 0. However, P1 could achieve a final payoff of 1 by playing t at any time. This contradicts Equation (3), and therefore P1 cannot play h forever.

Let $A(m)$ be the event that P1 plays a t by round m.

Claim 3.9 *There exists an m such that $Pr[A(m)] \geq 3/4$.*

Proof: Assume for all m that $\Pr[A(m)] < 3/4$. Then for all m, Pr[P1's first m moves are all h] $\geq 3/4$, and thus Pr[P1 always plays h] $\geq 3/4$. Therefore, against P2's T^* strategy, the expected final payoff of P1 is less than $0 \cdot 3/4 + 1 \cdot 1/4 = 1/4$. But this contradicts the above assumption that P1's strategy comes within 1/2 of the optimal strategy against P2's T^* strategy. However, the strategy $\sigma_1' = T^*$ gives P1 a final payoff of 1 against P2's T^*, which contradicts Equation (3). □ Claim 3.9

Now consider P1's strategy σ_1 against the strategy $\sigma_2 = T^m H^*$ for P2. P1 cannot differentiate between this strategy and T^* during the first m rounds, so by Claim 3.9, it is still true that $Pr[A(m)] \geq 3/4$. Therefore, P1's payoff is at most $1 \cdot 3/4 + 2 \cdot 1/4 = 5/4$. Against σ_2, P1 could have achieved a final payoff of 2 with the strategy H^*. Again, this contradicts Equation (3). □ Theorem 3.8

4 Conclusion and Open Questions

There has been much work in the area of bounded rationality in repeated matrix games (see for example [K, FW, PY]). However, to our knowledge, not much work has been done in the area of bounded rationality for general stochastic games. We would find it interesting to analyze bounded players in other stochastic games or prove general results about such games.

The Big Match is also a simple version of a bargaining game where two players try to decide how to share a limited resource. Very little work has progressed on looking at bounded players in these types of games.

Acknowledgments

We would like to thank Duke Whang and In-Koo Cho for very helpful discussions.

References

[BF] D. Blackwell and T. S. Ferguson, "The Big Match," Annals of Mathematical Statistics, Vol. 39, No. 1, pp. 159-163.

[FS] J.A. Filar and T.A. Schultz, "Bilinear Programming and Structured Stochastic Games," Journal of Optimization and Applications, April 1987, Vol. 53, No. 1, pp. 85-104.

[FW] L. Fortnow and D. Whang, "Optimality and Domination in Repeated Games with Bounded Players," Proc. 26th STOC, ACM, 1994, pp. 741-749.

[G] D. Gillette, "Stochastic Games with Zero Stop Probability," Contributions to the Theory of Games, Vol III. Princeton University Press, 1957, pp. 179-187.

[K] E. Kalai, "Bounded Rationality and Strategic Complexity in Repeated Games," Game Theory and Applications, Academic Press, Inc. 1990, pp. 131-157.

[MN] J.F. Mertens and A. Neyman, "Stochastic Games Have a Value," International Journal of Game Theory, 1981, Vol. 10, pp. 53-66.

[M] C.A. Monash, "Stochastic Games: The Minimax Theorem," PhD Thesis, Harvard University, Cambridge, Massachusetts, 1979.

[PTV] T. Parthasarathy, S.H. Tijs, O.J. Vrieze, "Stochastic Games with State Independent Transitions and Separable Rewards," Lecture Notes in Economics and Mathematical Systems, 1984, Vol. 226, pp. 262-271.

[PY] C.H. Papadimitriou and M. Yannakakis, "On Complexity as Bounded Rationality," Proc. 26th STOC, ACM, 1994, pp. 726-733.

[S] L.S. Shapley, "Stochastic Games," Proc. Nat. Acad. Sci., Vol 39, pp. 1095-1100.