PUC

Series: Monographs in Computer Science
and Computer Applications
Nº 2/74

THE USE OF LEARNING ALGORITHMS IN NON-MINIMAX
SOLUTION OF GAMES

by

Raimundo N. M. Chaves

Computer Science Department - Rio Datacenter

Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

THE USE OF LEARNING ALGORITHMS IN NON-MINIMAX SOLUTION OF GAMES

Raimundo N. M. Chaves
Master in Computer Science
Computer Science Department
PUC/RJ

Series Editor: S. M. dos Santos

1973

ABSTRACT

This paper deals with the problem of two-person, sero - sum games in the case that one of the players is using a strategy different from the usual von Neumann minimax stracegy.

The method is to use learning algorithms which are based on one player's estimate of the other's strategy after various repetitions of the same game. The algorithms attempt to let the first player take full advantage of the other's weaknesses.

Suppose at the K - th repetitions of the game the first and second player have apparent strategies of X(K) and Y(K) respectively. Then the strategy of the first player for the next repetitions is determined by the difference equation.

$$X(K+1) = X(K) - \gamma(K+1)\,(X(K) - f(Y(K))).$$

f(Y(K)) represents the optimal pure response of the first player to the strategy Y(K). γ (K+1) is a parameter to be chosen to assign weights to different strategies.

Various models of this situation are studied by varying the different parameters of the equation. Empirical tests of the results are made using a 3 x 3 payoff matrix which has no saddle point and a unique minimax solution.

1. INTRODUCTION

One of the difficulties in the application of game theory has been the restriction of game solutions to games with complete information. It is generally assumed that each player is in possession of all possible information about the structure of the game and about the expected behavior of his opponents.

A general discussion of these limitations is found in Harsanyi[1]. In this paper we will only attempt to modify the restrictions dealing with the expected behavior of opponents.

We will assume some familiarity with classical twoperson, zero-sum game theory, as has been developed by von Neumann and others. A good exposition of the subject is given in McMinsey[2]. Suppose that we are given a two-person, zero-sum game with m x k payoff matrix $A=(a_{ij})$. Each line of the matrix is a pure strategy for P1, and each column a pure strategy for P2. The usual solution is the well-know minimax solution given by (generally mixed) strategies X^* for P1 and Y^* for P2 plus the value $V^* = E(X^*, Y^*)$ of the game. These satisfy the relation, that for all possible strategies $X=(x_1,\ldots,x_m)$ and $Y=(y_1,\ldots,y_k)$ we have

$$E(X,Y^*) \leq E(X^*,Y)$$

where the mathematical expectation of the payoff for P1 is given by

$$E(X,Y) = \sum_{i=1}^{m} \sum_{j=1}^{k} a_{ij} x_i y_j$$

The use of the minimax strategy X^* by P1 will guarantee that his expected payment will be at least v^* no matter what his opponent does. If his opponent also plays a minimax strategy, then X^* will indeed be an optimal strategy for P1.

2. PROBLEM STATEMENT

We will be interested in seeing what happens if P2 does not play according to a minimax strategy Y^*. Suppose P2 is playing whith an apparent strategy $\bar{Y} \neq Y^*$. In general $E(X^*,\bar{Y})$ v^*, but there may be different strategies X for P1 such that $E(X,\bar{Y}) > E(X^*,\bar{Y})$. In other words, there may be strategies for P1 which will allow him to fully capitalize on the "mistake" of P2. If the strategy $\bar{Y}$ of P2 is know with certainty, then it is a trivial mathematical problem to find a strategy $\bar{X}$ which will maximize $\{E(X,Y^*)\}$. Unfortunately, due to our assumption of incomplete information, we will generally not know $\bar{Y}$. If the game is to be played only once, then there are two possibilities - either (a) we know nothing about P2, so we should play X^* or (b) we might be able to obtain psychological information about P2 which would enable us to estimate his strategy $\bar{Y}$.

Mathematically the problem becomes more interesting if the game is repeated many times. We will then be in a position to learn $\bar{Y}$ from the relative frequencies of the past plays of P2. However, in order not to place too much confidence in premature results, we will generally give some weigh to use of the minimax strategy X^*.

3. SOLUTION METHOD

P1 will be simulated by a computer program which can find the minimax solution of a game, as well as having several learning algorithms on the basis of which it willattempt to improve upon the original minimax solution. We intend to have an interactive program and have P2 be a human player, who may, but will generally not,know game theory. Due to present lack of interactive facilities P2 will

also be simulated by the computer, playing with either randomly selected or predetermined strategies not "know" to P1.
It will be assumed that P1 only knowledge of the minimax solution of the game and of the past history of his own choices and those of P2.

4. GENERAL MODEL

Let us denote by X(n) the (mixed) strategy which P1 uses for the n - th repetition of the game, and sililarly Y(n)for P2. For a given strategy Y of P2 let us denote by $U^i(Y)$ the optimal pure strategy response to Y. That is to say, if P2 uses Y then U^i (Y) guarantees P1 a payment at least as large as any of the other m-1 pure strategies. To avoid ambiguity, if there are several pure strategies which are maximal pure responses to Y then we let $U^i(Y)$ be the first of these relative to the ordering of the numbers of the lines of the matrix. Similarly for strategy X by P1, we let $T^j(X)$ be the optimal pure response of P2 to X.

Tsypkin[3] derives a class of algorithms for learning the minimax solution of a game. These are generally of the following form: $(n \geq 1)$

$$\begin{aligned} X(n) &= X(n-1) - \gamma_1(n) \ (X(n-1) - U^i(Y(n-1))) \\ Y(n) &= Y(n-1) - \gamma_2(n) \ (Y(n-1) - T^j(X(n-1))) \end{aligned} \qquad (1)$$

$\gamma_1(n)$, $\gamma_2(n)$ are numerical functions of n, which must be specified, for a particular algorithm. In addition, in order to begin the algorithm we must specify some initial strategies X(0) and Y(0). The algorithms are intended to be used in an artificial situation where

P1 and P2 both selected their strategies at each repetition according to the algorithm. The sequences X(n) and Y (n) will then converse to a minimax solution (X^*, Y^*) of the game.

Our situation is somewhat different. Since we only control the behavior of P1 and not of P2, it makes no sence to include the equation for Y(n). Thus the sequence Y(0), Y(1), ... , Y(n-1) will not be developed according to the algorithm. However, we can estimate the apparent strategy of P2 at the n-th repetition from the knowledge of the relative frequencies of this first n-1 choices. Since P1 already knows X^* before the begining of the games we will usually let $X(0) = X^*$. In the sequel we will experiment with various function (n). Our algorithm can be expressed as

$$X(0) = X^*$$
$$X(1) = X^* - \gamma(1)\,(X^* - U^i(Y(0)))$$
$$X(2) = X(1) - \gamma(2)(X(1) - U^i(Y(1)))$$

.

.

.

where we can now substitute X(1) by the previously found value, etc. In the absence of a priori information about P2, Y(0) must necessarily be arbitrary. Sometimes we will assume $Y(0) = (1/k, \ldots, 1/k)$.

4.1. SPECIFIC MODELS

We have experimented with three basic models which we will call MB1, MB2, MB3. These are determined by the algorithm (1) and by the choice of γ (n).

Let us discuss briefly the form which each of these models takes, as well as some possible modifications in the algorithm, which may be used in conjuction with each model. Then an attempt is made to compare the perfonance of the different models on the basis of some experimental results.

MB1 is characterized by the choice of $\gamma(n) = 1/c$ where c is a constant positive integer with $c \geq 2$.

In particular, we have experimented with c=2. It is seen that in this model the algorithm reduces to

$$X(n) = (c-1)^n / c^n \cdot X(0) + (c-1)^{n-1} / c^n \cdot U^i (Y(0)) +$$

$$(c-1)^{n-2} / c^{n-1} \cdot U^i(Y(L)) + \ldots + (c-1)^0 / c^1 \cdot U^i(Y(n-1)).$$

In otherswords the algorithm is of the form

$$X(n) = b \cdot X(0) + a_0 \cdot U^i (Y(0)) + \ldots + a_{n-1} \cdot U^i(Y(n-1)),$$

whith $a_0 < a_1 < a_{n-1}$. Thus a larger weight is given to the more recent estimates of the strategy of P2. This seems reasonable, since our estimate of his strategy should normally improve with a large number of repetitions. As we will see, however, this algorithm will not generally converge to a minimax solution of the game even when used by both players.

MB2 uses the parameter $\gamma(n) = 1/n$, which makes the algorithm reduce to the classical Brown method for approximating the value of a rectangular game (see McKinsey[2]). The formula for this model can be reduced to

$$X(n) = 1/n. \sum_{k=0}^{n-1} U^i(Y(K)).$$

In general, this model assigns the same weight to all of the estimates of P2's strategy.

Finally MB3 uses values of $\gamma(n)$ determined by

$\gamma(n) = r_n / \sum_{S=1}^{n} r_s$ where r_i is the total number of times that the pure strategy of P2 at the i - th repetition has been used in all of the i repetitions. MB3 was used by Amvrosienko to aproximate the solution of a game (see Tsypkin[3]). For MB3 the formula reduces to

$$X(n) = (1 / \sum_{S=1}^{n} r_s). \sum_{S=1}^{n} r_s \; U^i(Y(s-1)).$$

The weight given to each factor depends on the number of times that the same choice has been made before by P2.
In general, with this model there is a tendendy to assign a larger weight to more recent information.

Before discussing some empirical results we wish to mention briefly two modifications which may be made in the three models discussed above. This will give a total of nine models that will be compared, i.e. each of the basic models without modification and each ba basic model with each of the modifications.

The first modification M1 involves using $U^i(Y(m))$ in the basic algorithms where $U^i(Y(m))$ represents a function of the apparent strategy in the last m games instead of simply the last game. The number m amy be chosen as a function of n.

If P2 has certain types of well-defined behavior, then we are sometimes interested in accelerating the convergence of the sequence X(n). Note, however, that the sequence does not generally converge, depending on the behavior of P2. If it does converge then we may increase the rate of convergence by using the following recursion (M2):

$$X(n) = X(n-1) - \gamma(n) \; (X'(n-1) - U^i(Y(n-1)))$$

where $X'(n-1) = (1-\alpha).\, X(n-1) + \alpha.X(n-2)$

The restriction upon α will be given below. Generally the convergence will be accelerated in this manner if the sequence X(n) is converging to a limit in an oscillating fashion.

If is calculated according to the formula

$$\alpha = (X(n) - X(n-1)) \;/\; (X(n) - 2.\, X(n-1) + X(n-2))$$

Then we obtain the Aitken formula for accelarating the convergence of a sequence:

$$X'(n) = X(n-2) - (X(n-1) - X(n-2))^2 / \; (X(n)-2.X(n-1)+X(n-2))$$

5. TESTS OF THE MODELS

For each of the nine models we will use four tests in order to compare the relative effectiveness of each model in various situations. There is no attempt made yet to reach definitive conclusions, but merely to study some computational results from different points of view. At present the choice of models and tests in somewhat arbitrary. The tests to be considered here are:

TEST 1 - Here it is supposed that P2 is using a strategy determined by the same algorithm as the strategy of P1. In other words the original models (1) of two simultaneous recursions are used with $\gamma_1(n) = \gamma_2(n)$. The test is to check for convergence of the sequence of payoffs v(n) to the value v^* of the game, and to test for the convergence of X(n) and Y(n) to a pair of optimal strategies (X^*,Y^*).

TEST 2 - In this test P2 will use the strategy Y^* for all repetitions of the game. Each model will be tested for the convergence of X(n) to the optimal strategy X^*.

TEST 3 - In this test P2 will be playing with a pre-determined mixed strategy Y, but instead of making random choices according to Y he will simply choose the first column r_1 times, then the second column r_2 times, etc. Here we are interested in the sensitivity of each model to apparent changes of strategy P2.
The model which is most sensitive will, in this situation, obtain a higher average payoff for P1.

TEST 4 – For this test it is assumed that P1 is playing a "good" mixed strategy, but not quite optimal. In general, the model should be able to obtain an average payoff which is more than the value v^* of the game.

It is hoped that at least the three last tests simulate behavior which would be reasonably commom in real opponents. First, there is the knowledgeable player who has learned game theory. Second we have the type of player who tries a pure strategy a few times, sees that it is beginning to have bad consequences, then changes to another strategy until that too has bad consequences. Finally there is the intelligent opponent who knows he must use a mixed strategy in order to keep the other player guessing. He is able to do a reasonable analysis of the game, but due to lack of training cannot calculate the optimal strategy.

6. PROGRAM STRUCTURE

The program written in PL/1 follows the basic flow chart of Figure 1. It consists basically of six procedures as follows:

1. SIMPLEX – This is used to calculate the von Neumann solution of the game. The method used here is found in Charnes[4] who reduces the problem to the following pair of linear programming problems:

Primal	Dual
max V	min w
such that	such that
$X^tA \geq vE_k$	$AY \leq WE_m$
$X^tE_m = 1$	$YE_k = 1$
$X \geq 0$	$Y \geq 0$

where E_n is the n - component vector with each component equal to 1.

The solution used is a revised simplex method found in Lasdon[5].

2. RANDU - This is used to generate a random integer i in the interval $1 \leq i \leq I$. The method used is that of multiplicative congruence (see Naylor[6]) defined by the relation

$$n_{i+1} = a.n_i \pmod m$$

for generating pseudo - random numbers. The initial value n_0 is obtained from the clock of the computer.

3. EST-PURA - This procedure is used to determine an integer i, $1 \leq i \leq I$ chosen "pseudo-randomly" in accordance with the probability distribution X.

4. M1 - This uses an I x M matrix, where I is the dimension of $U^i(Y)$ and M is the number of vector desired to calculate the weighted average

$$U^i(Y(n)) = 1/M \sum_{m=n-M}^{n} U^i(Y(m))$$

5. M2 - This is used to calculate , $X'(k)$ as in the modification M2.

6. LEARN - This is the principal procedure of the program. It first activates EST-PURA whith X^* to determine the first choice X(O). Then it calculates X(k) successively according to the model used until there is a flag, at which time it will calculate the statistics of the outcomes. (Fig.2)

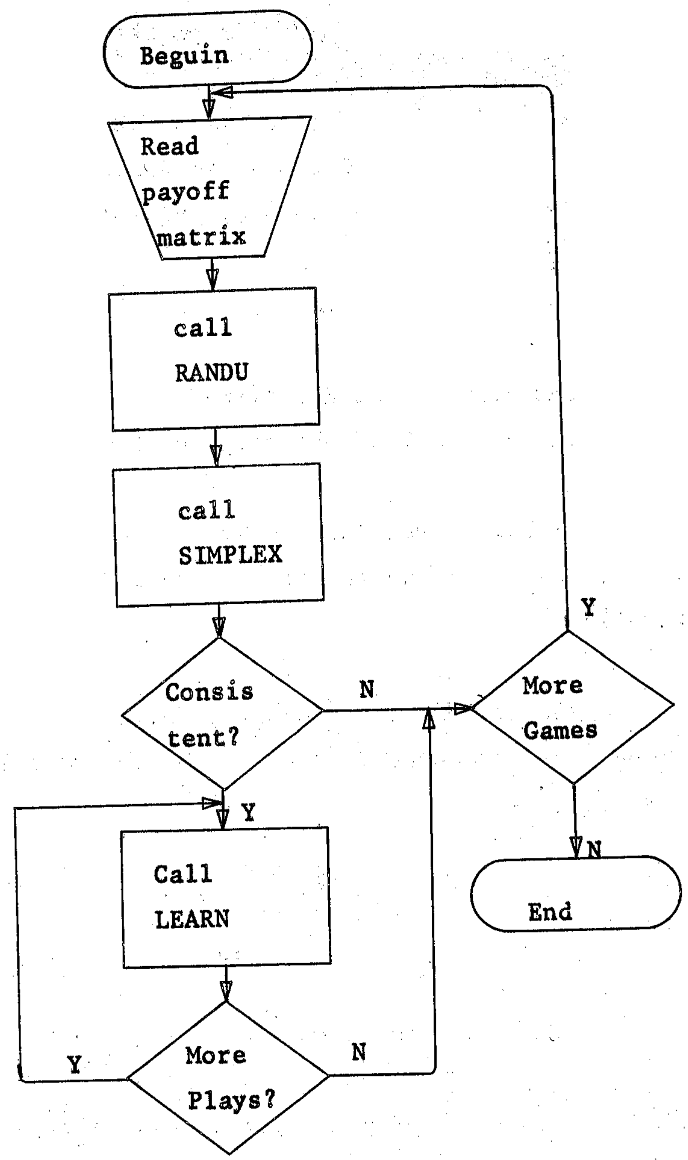

Fig. 1: Simplified flowchart of the general program

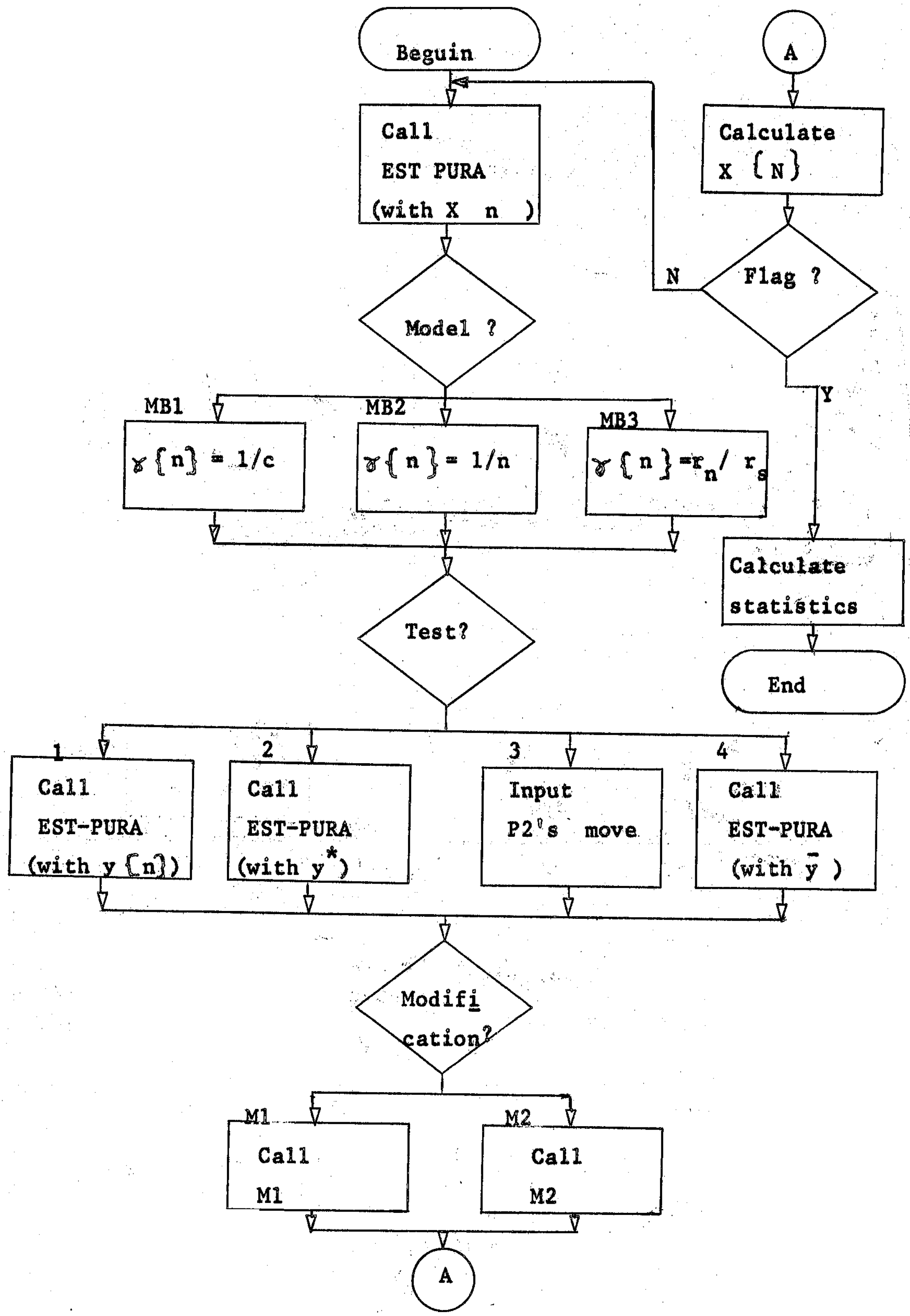

Fig.2 : Simplified flowchart of the LEARN procedure

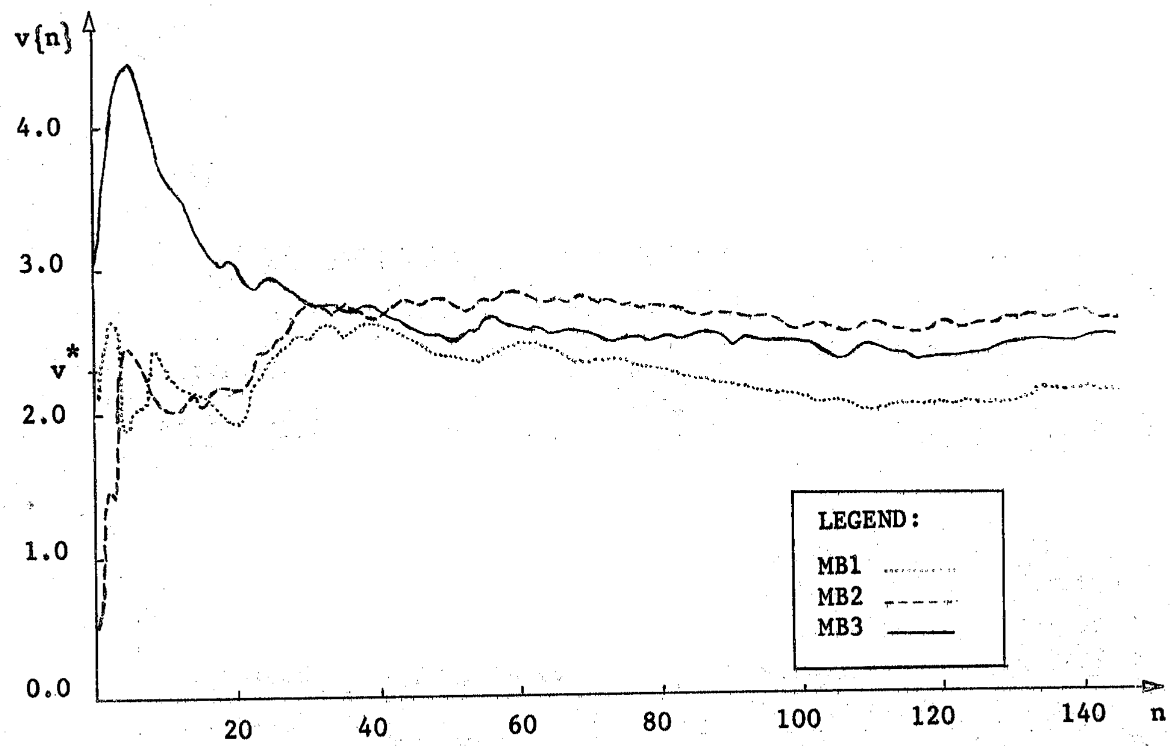

Fig.3: Average payoff to P1 after n repetitions. Results of test 2.

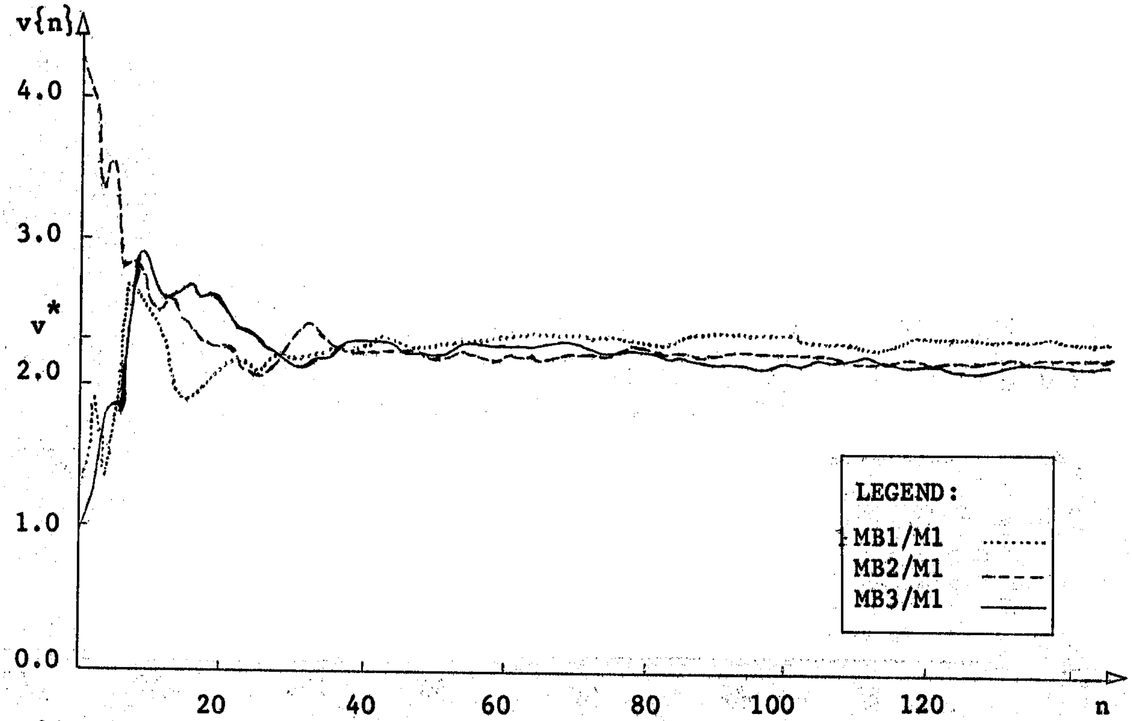

Fig. 4: Average payoff to P1 after n repetitions. Results of test 2.

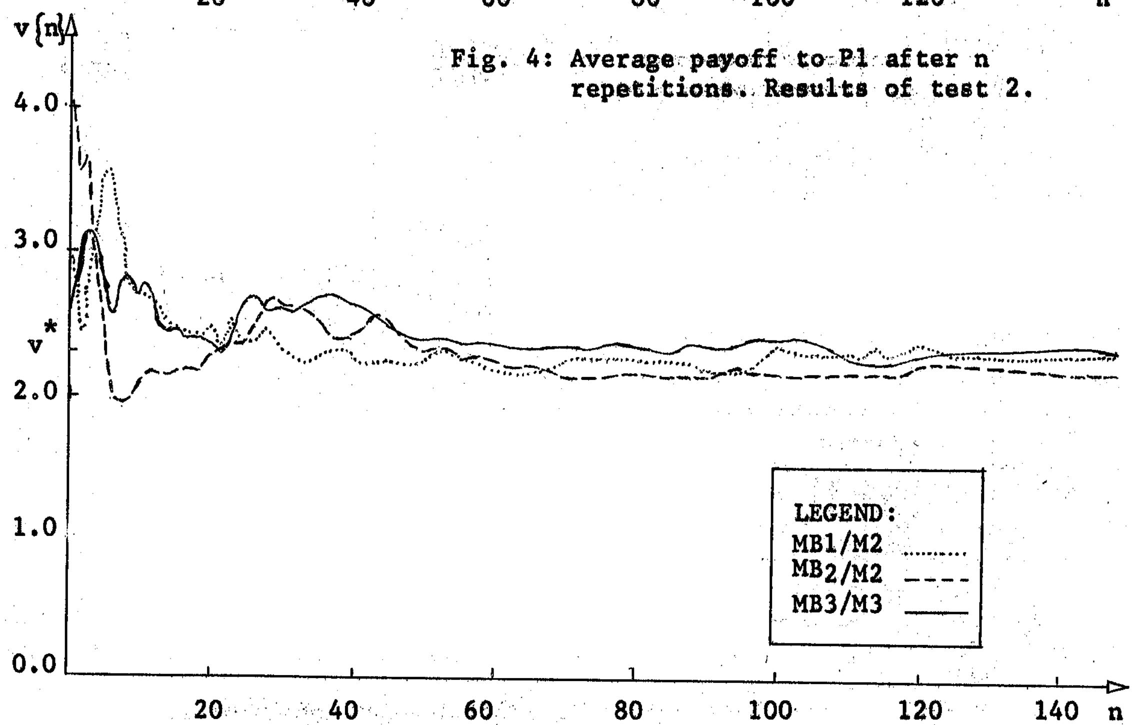

Fig.5: Average payoff to P1 after n repetitions. Results of test 2.

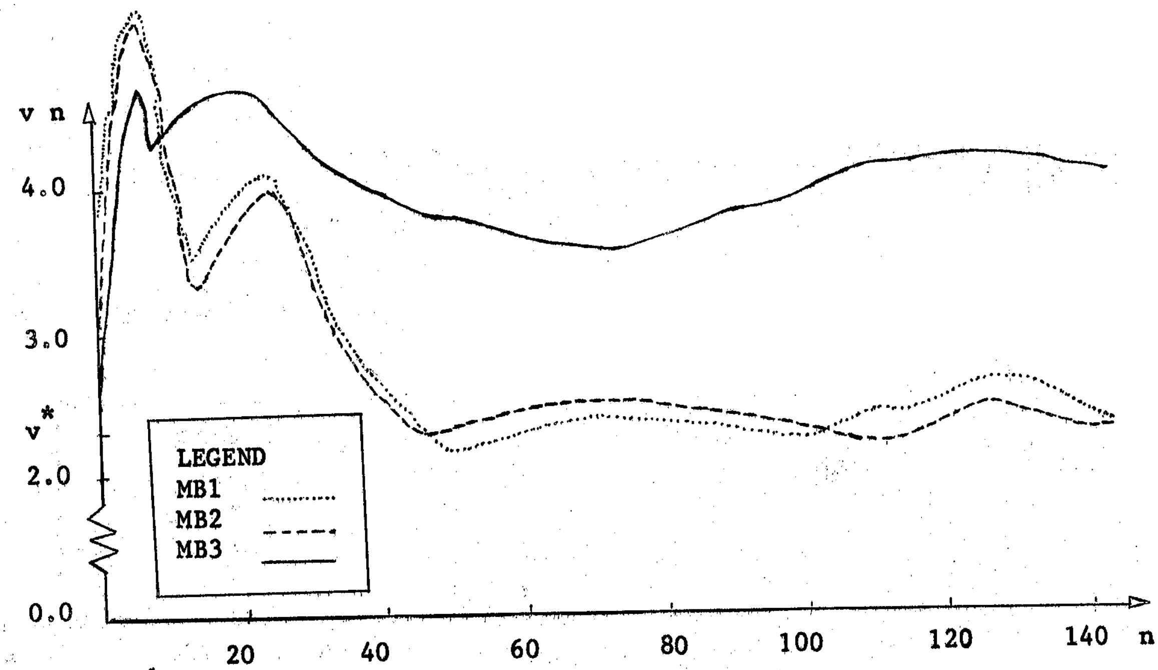

Fig.6: Average payoff to P1 after n repetitions. Results of test 3.

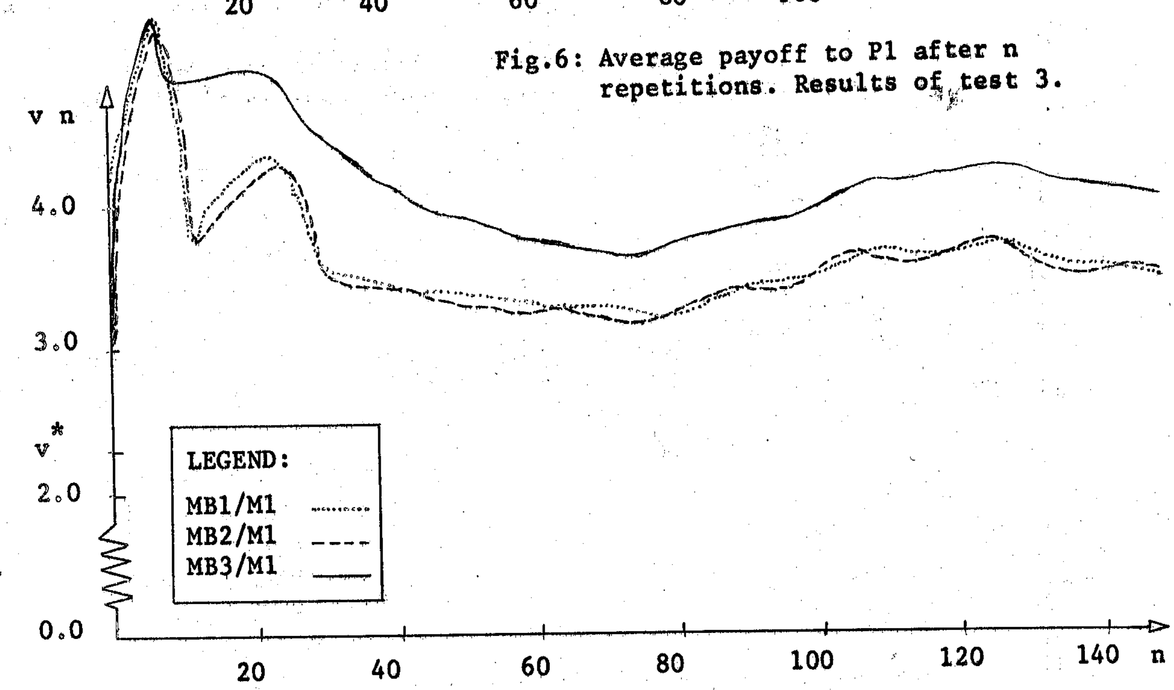

Fig. 7 : Average payoff to P1 after n repetitions. Results of test 3

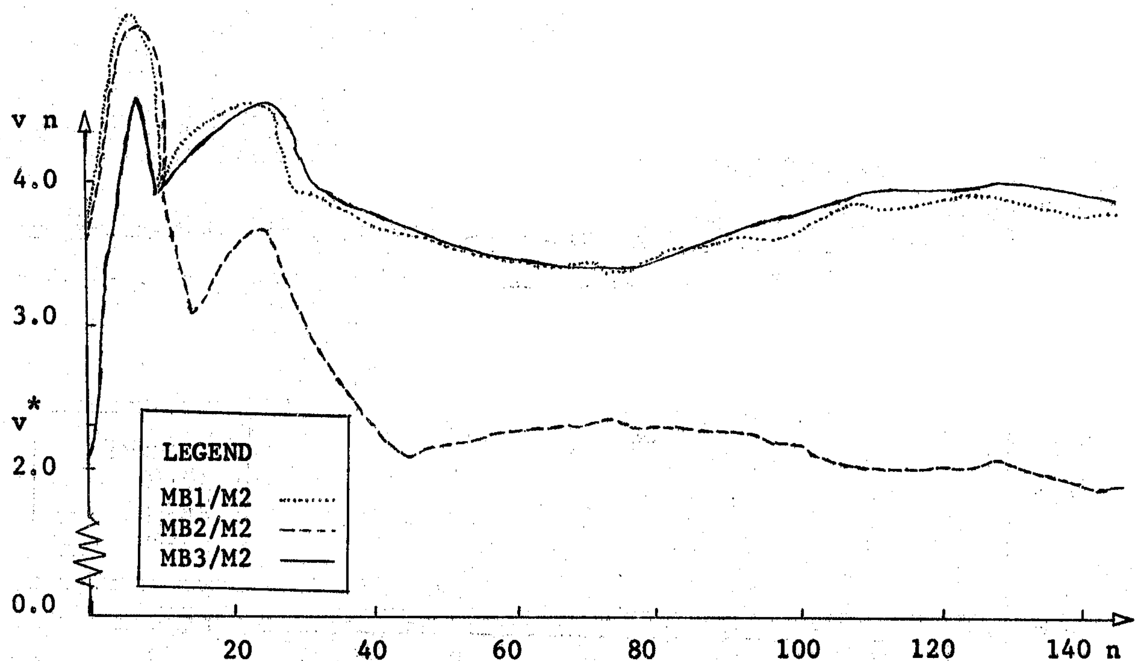

Fig.8: Average payoff to P1 after n repetitions. Results of test 3.

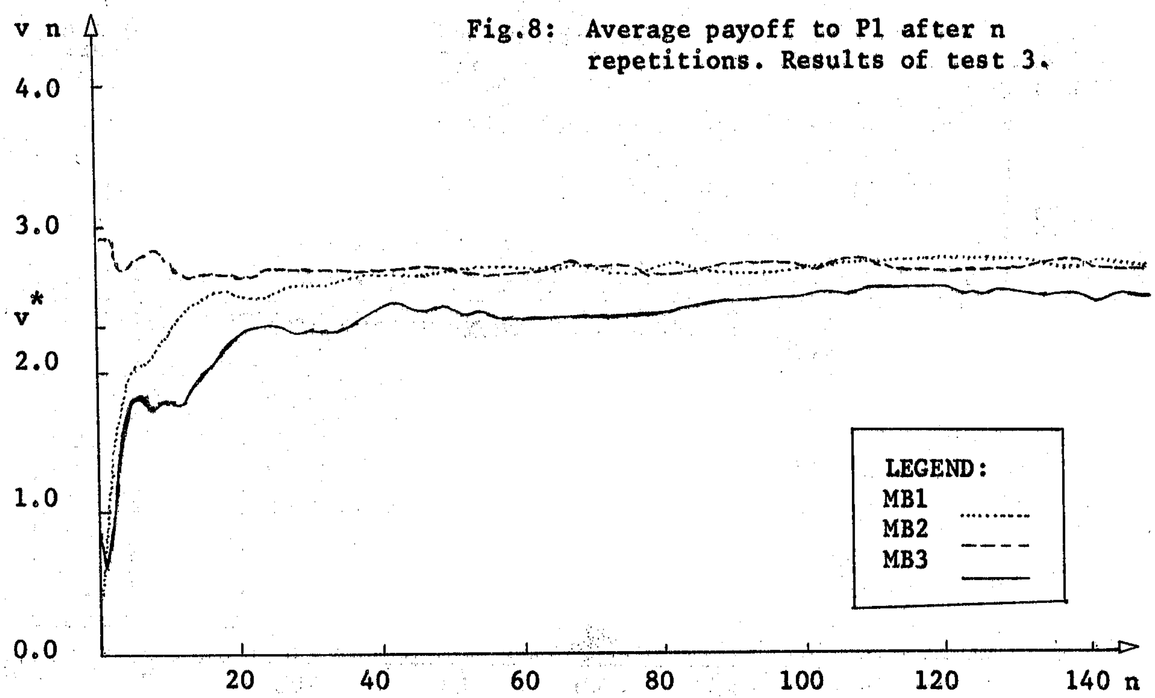

Fig. 9: Average payoff to P1 after n repetitions. Results of test 4.

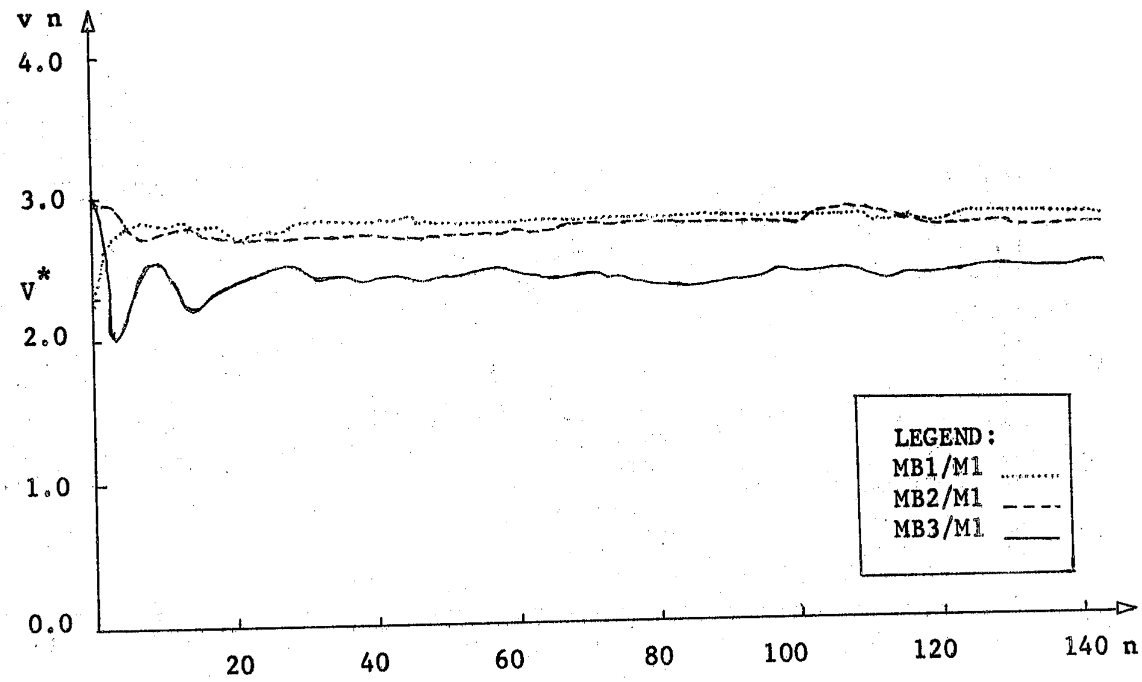

Fig.10 Average payoff to P1 after n repetitions. Results of test 4.

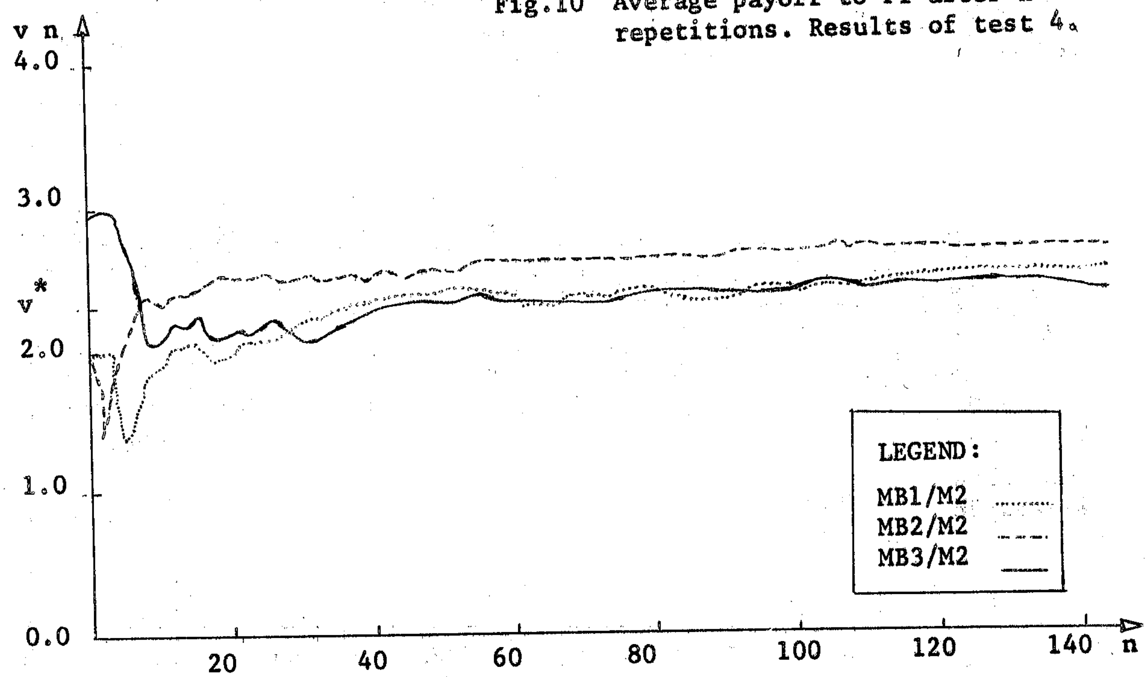

Fig.11: Average payoff to P1 after n repetitions. Results of test 4.

7. DATE USED

In order to apply the four tests mentioned above we used the payoff matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 1 & 2 \\ 3 & 5 & 0 \end{bmatrix}$$

This game has the unique solution

$$X^* = (.639 \quad .194 \quad .167)$$

$$Y^* = \left[.167 \quad .361 \quad .472\right]$$

$$V^* = 2.3056$$

A game with a unique solution was chosen because the amount of deviation from an optimal strategy is seen more clearly.

The following conditions were imposed on the experiments:

1. In the case of MB1,MB1/M1, MBL/M2 the value of c was chosen to be c=2
2. The number of repetitions for all tests was fixed at 200,
3. In the modification M1, the last 10 plays were used.
4. In the case of test 4, it was assumed that P2 would play with a strategy (0 .25 .75).
5. For the modification M2 must satisfy the restriction that it make X'(n-1) a legitimate strategy (i.e. X'(n-1) 0).

8. TENTATIVE CONCLUSIONS

The results of the tests (in the form of the payoff to P1) are given in the graphs of Fig.3 to Fig.11. On the basis of the figures and tables we may arrive at some very tentative conclusions:

a) Table 1 indicates that in the first test X(n), Y(n) converged to the minimax solution in the case of MB2 and MB3, but not in the case of MB1.

b) The results of the second test show that all of the models converge to v under the condition of the test. This is seen in Fig.3 to Fig. 5 and in Table 2.

c) With the condition of test 3, we can see in Fig. 6 to Fig. 8 and Table 3 that MB1 gives appreciably better results than either MB2 or MB3. Also MB1 gives better results than X^* would have given under the same circumstances.

d) An interesting conclusion is that it appears that the modification M1 appreciably increases P1's payoff. Thus it seems that introducing memory into the system improves the results. See Figures 4,7,10.

e) Modification M2, which was introduced to accelerate convergence, appears to have no effect on the average outcome. This can be seen in Figures 5,8 and 11.

TABLE 1:

model	nº of plays	P1's current strategy			average payoff to P1	P2's estimated strategy			average payoff to P2
	10	.580	.135	.285	2.729	.064	.143	.793	2.010
	20	.400	.533	.067	3.467	.267	.533	.200	1.133
M	40	.400	.533	.067	3.467	.267	.533	.200	1.667
	60	.400	.533	.067	3.467	.267	.533	.200	1.667
B	80	.400	.533	.067	3.467	.267	.533	.200	1.667
	100	.400	.533	.067	3.467	.267	.533	.200	1.667
1	150	.600	.133	.267	2.733	.067	.133	.800	2.067
	200	.400	.533	.067	3.467	.267	.533	.200	2.067
	10	.389	.500	.111	2.550	.100	.450	.450	1.833
	20	.605	.237	.158	2.475	.050	.425	.525	2.237
M	40	.615	.231	.154	2.337	.150	.362	.487	2.231
	60	.636	.203	.161	2.325	.150	.375	.475	2.280
B	80	.570	.253	.177	2.344	.162	.331	.506	2.215
	100	.626	.202	.172	2.365	.130	.375	.495	2.283
2	150	.597	.235	.168	2.377	.167	.290	.543	2.262
	200	.648	.181	.171	2.352	.175	.347	.477	2.246
	10	.619	.190	.190	2.773	.227	.136	.636	2.238
	20	.600	.144	.256	2.568	.062	.309	.630	2.089
M	40	.584	.243	.173	2.573	.104	.218	.678	2.238
	60	.503	.317	.179	2.345	.161	.373	.467	2.145
B	80	.620	.167	.213	2.451	.085	.379	.536	2.194
	100	.526	.291	.184	2.340	.149	.363	.489	2.158
3	150	.606	.235	.159	2.355	.137	.370	.493	2.242
	200	.566	.273	.161	2.336	.161	.370	.471	2.211

TABLE 2:

model	nº of plays	without modification: P2's estimated strategy	without modification: average payoff to P1	with modification 1: P2's estimated strategy	with modification 1: average payoff to P1	with modification 2: P2's estimated strategy	with modification 2: average payoff to P1
	10	.31 .06 .63	3.800	.95 .00 .95	3.000	.25 .50 .25	2.600
	20	.64 .00 .36	3.550	.00 .10 .90	2.600	.01 .56 .43	2.350
M	40	.50 .40 .10	2.775	.03 .02 .95	2.275	.57 .04 .39	2.600
	60	.02 .36 .62	2.600	.28 .65 .07	2.250	.26 .64 .10	2.333
B	80	.63 .25 .12	2.512	.52 .23 .25	2.225	.56 .04 .40	2.337
	100	.06 .41 .53	2.450	.00 .02 .98	2.150	.63 .12 .25	2.390
1	150	.63 .02 .35	2.507	.13 .57 .30	2.133	.25 .59 .16	2.307
	200	.25 .64 .11	2.457	.00 .02 .98	2.251	.28 .51 .21	2.236
	10	.40 .30 .30	2.300	.40 .30 .30	2.500	.00 .49 .51	2.700
	20	.20 .35 .45	1.850	.00 .30 .70	2.100	.06 .70 .24	2.550
M	40	.30 .28 .42	2.625	.10 .50 .40	2.300	.00 .80 .20	2.300
	60	.22 .28 .50	2.385	.10 .20 .70	2.383	.06 .40 .54	2.233
B	80	.19 .32 .49	2.275	.20 .40 .40	2.375	.04 .09 .87	2.250
	100	.18 .37 .45	2.110	.20 .20 .50	2.380	.01 .88 .11	2.310
2	150	.15 .37 .48	2.067	.00 .50 .50	2.380	.50 .06 .44	2.333
	200	.14 .38 .48	2.126	.10 .70 .20	2.387	.00 .22 .78	2.427
	10	.21 .04 .75	2.000	.04 .58 .38	2.800	.06 .15 .79	2.100
	20	.14 .03 .83	2.150	.04 .38 .58	2.300	.00 .96 .04	2.350
M	40	.29 .11 .60	2.600	.04 .21 .75	2.275	.04 .74 .22	2.400
	60	.36 .24 .40	2.783	.46 .27 .27	2.283	.07 .57 .36	2.300
B	80	.29 .34 .37	2.662	.00 .82 .18	2.313	.01 .11 .88	2.212
	100	.23 .36 .36	2.550	.08 .00 .92	2.320	.03 .18 .79	2.200
3	150	.20 .31 .40	2.520	.00 .18 .82	2.313	.01 .03 .96	2.253
	200	.16 .34 .50	2.427	.00 .82 .18	2.407	.00 .07 .93	2.312

TABLE 3:

model	nº of plays	whithout modification		with modification 1		with modification 2	
		P2's estimated strategy	average payoff to P'	P2's estimate strategy	average payoff to P1	P2's estimated strategy	average payoff to P1
	10	.25 .75 .00	4.400	.25 .75 .00	4.900	.50 .50 .00	4.000
	20	.00 100 .00	4.700	.00 100 .00	4.950	.00 100 .00	4.500
M	40	.00 .00 100	4.075	.00 .00 100	4.200	.00 .00 100	3.825
	60	.00 .00 100	3.717	.00 .00 100	3.800	.00 .00 100	3.550
B	80	.00 .99 .01	3.675	.00 .99 .01	3.737	.00 100 .00	3.550
	100	.99 .01 .00	3.950	.99 .01 .00	4.000	100 .00 .00	3.850
1	150	.00 .00 100	4.027	.00 .00 100	4.060	.00 .00 100	3.893
	200	.00 .00 100	4.025	.00 .00 100	4.050	.00 .00 100	3.925
	10	.80 .20 .00	4.500	.80 .20 .00	4.500	.90 .10 .00	4.500
	20	.40 .60 .00	3.950	.00 100 .00	4.150	.90 .10 .00	4.550
M	40	.20 .45 .35	2.725	.00 .00 100	3.425	.00 .01 .99	3.850
	60	.14 .30 .56	2.317	.00 .00 100	3.283	.00 .00 100	3.567
B	80	.10 .31 .59	2.400	.70 .30 .00	3.162	.00 .88 .12	3.525
	100	.15 .38 .47	2.270	.70 .30 .00	3.420	.88 .12 .00	3.770
2	150	.17 .37 .46	2.280	.00 .00 100	3.467	.00 .00 100	3.813
	200	.17 .37 .46	2.295	.00 .00 100	3.440	.00 .00 100	3.800
	10	.92 .08 .00	4.500	.92 .08 .00	4.500	.95 .05 .00	4.500
	20	.32 .68 .00	3.750	.00 100 .00	4.150	.00 100 .00	3.550
M	40	.11 .55 .34	2.625	.00 .00 100	3.425	.00 .66 .34	2.525
	60	.05 .21 .74	2.450	.00 .00 100	3.283	.00 .10 .90	2.450
B	80	.02 .22 .76	2.500	.00 .82 .18	3.162	.00 .00 100	2.383
	100	.06 .37 .57	2.330	.82 .18 .00	3.420	.04 .29 .67	2.290
3	150	.08 .37 .55	2.300	.00 .00 100	3.467	.00 .00 100	2.013
	200	.08 .36 .56	2.340	.00 .00 100	3.445	.00 .00 100	2.080

TABLE 4:

model	nº of plays	without modification: P2's estimated strategy	without modification: average payoff to P1	with modification 1: P2's estimated strategy	with modification 1: average payoff to P1	with modification 2: P2's estimated strategy	with modification 2: average payoff to P1
	10	.00 .56 .44	1.800	.00 .00 100	2.600	.00 .23 .77	1.800
	20	.00 .50 .50	2.200	.00 .01 .99	2.400	.00 .28 .72	1.900
M	40	.00 .88 .12	2.450	.00 .06 .94	2.450	.00 .03 .97	2.325
	60	.00 .31 .79	2.317	.00 .00 100	2.533	.00 .27 .73	2.367
B	80	.00 .03 .97	2.362	.00 .30 .70	2.425	.00 .02 .98	2.400
	100	.00 .70 .30	2.440	.00 .22 .78	2.460	.00 .01 .99	2.420
1	150	.00 .63 .37	2.427	.00 .14 .86	2.467	.00 .09 .91	2.540
	200	.00 .26 .74	2.452	.00 .25 .75	2.528	.00 .13 .87	2.528
	10	.00 .30 .70	2.300	.00 .10 .90	2.800	.00 .00 100	2.100
	20	.00 .20 .80	2.600	.00 .30 .70	2.750	.00 .33 .67	2.150
M	40	.00 .18 .82	2.725	.00 .10 .90	2.850	.00 .00 100	2.300
	60	.00 .17 .83	2.767	.00 .20 .80	2.800	.00 .54 .46	2.367
B	80	.00 .22 .78	2.725	.00 .30 .70	2.800	.00 .00 100	2.400
	100	.00 .23 .77	2.730	.00 .10 .90	2.810	.00 .00 100	2.440
2	150	.00 .25 .75	2.727	.00 .10 .90	2.793	.00 .00 100	2.453
	200	.00 .24 .76	2.744	.00 .10. 90	2.714	.00 .00 100	2.472
	10	.00 .02 .98	2.900	.00 .08 .92	2.800	.00 .05 .95	2.400
	20	.00 .11 .89	2.750	.00 .18 .82	2.750	.00 .01 .99	2.600
M	40	.00 .11 .89	2.750	.00 .18 .82	2.750	.00 .05 .95	2.575
	60	.00 .12 .88	2.733	.00 .00 100	2.733	.00 .00 100	2.650
B	80	.00 .13 .87	2.725	.00 .08 .92	2.813	.00 .01 .99	2.662
	100	.00 .16 .84	2.700	.00 .18 .82	2.800	.00 .00 100	2.720
3	150	.00 .16 .84	2.700	.00 .02 .98	2.800	.00 .00 100	2.700
	200	.00 .12 .88	2.729	.00 .18 .82	2.764	.00 .00 100	2.709

BIBLIOGRAFHY:

1. HARSANYI , J.C., "Games with incomplete information played by "Bayesian" players. I" , Management Science, vol.14, nº 3, 159 - 182 , 1967.

2. McKINSEY, J.C.C., Introduccion a la Teoria Matematica de los Juegos, Madrid, Aguilar, 1967

3. TSYPKIN, Y.Z., Adaptation and Learning in Automatic Systems, New York, Academic Press, 1971

4. CHARNES, A., et alli, "Chance - sonstrained games with partially controlable strategies", Operations Research 16, 142 -149, 1968.

5. LASDON, L.S. Optimization Theory for Large Systems, New York, MacMillan , 1970.

6. NAYLOR, T.H., et alli, Computer Simulation Techniques, New York, Wiley, 1968.