CHAPTER XXIV

ITERATIVE SOLUTION OF GAMES BY FICTITIOUS PLAY

BY GEORGE W. BROWN

It is the purpose of this chapter to describe and to discuss briefly a simple iterative method for approximating to solutions of discrete zero-sum games. This method is related to some particular systems of differential equations, which have been considered, along with some other systems of differential equations whose steady state solutions correspond to solutions of a game. Some of these and similar dynamical systems have been considered independently by von Neumann and will be treated in a joint paper [1950].

The iterative method in question can be loosely characterized by the fact that it rests on the traditional statistician's philosophy of basing future decisions on the relevant past history. Visualize two statisticians, perhaps ignorant of min-max theory, playing many plays of the same discrete zero-sum game. One might naturally expect a statistician to keep track of the opponent's past plays and, in the absence of a more sophisticated calculation, perhaps to choose at each play the optimum pure strategy against the mixture represented by all the opponent's past plays.

For calculation purposes the rule used here is that strategies will be named in turn for each side, choosing at each turn a pure strategy which is optimal against the cumulated history of the opponent's plays to date. Stated algebraically, let A_{ij} represent the matrix of payments from player 2 to player 1, let i_n and j_n be the nth choices of pure strategies for the two sides, let $\xi_i^{(n)}$ and $\eta_j^{(n)}$ be the relative frequencies of strategies i and j in $(i_1, i_2, \cdots, i_n)$ and $(j_1, j_2, \cdots, j_n)$, respectively; then the rule adopted is that j_n should minimize $\sum_i \xi_i^{(n)} A_{ij}$ and i_{n+1} should maximize $\sum_j A_{ij}\eta_j^{(n)}$. If i_1 is specified arbitrarily, this defines the sequence $i_1, j_1, i_2, j_2, \cdots$ recursively, except for possible multiple extrema, which can be handled by any convenient rule, for example, by ordering the strategies for each player. Letting $\underline{V}_n = \min_j \sum \xi_i^{(n)} A_{ij}$ and $\overline{V}_n = \max_i \sum A_{ij}\xi_j^{(n)}$, it is readily seen that $\underline{V}_n \leqq V \leqq \overline{V}_n$, where V is the value of the game. If the method is to be successful it is to be hoped that $\underline{V}_n$ and $\overline{V}_n$ converge to V or, at least, that their limits superior and inferior, respectively, are equal. The mixtures $\{\xi_i^{(n)}\}$ and $\{\eta_j^{(n)}\}$ represent mixed strategies, and the corresponding $\underline{V}_n$ and $\overline{V}_n$ are the most favorable payoff that either player could insure by the use of the particular mixture indicated.

It should be clearly pointed out that practically no nontrivial properties of this iterative scheme have been mathematically established.[1] All that is rigorously proved so far is that if the $\overline{V}_n$ and $\underline{V}_n$ converge at all they must converge to V. This is comforting, of course, but far from sufficient to justify the method. Some support can be gained from the relation the method bears, as a difference equation, to a set of differential equations whose convergence can be shown to be like $1/t$, where t corresponds to n in the discrete version. In the system of differential equations the convergence rests on the fact that $t\overline{V}(t)$ and $t\underline{V}(t)$ maintain a constant difference between them. Further empirical support can be drawn from the experience gained in using the method on a number of different examples, of dimensions up to the now-famous diet problem (which reduces, in one form, to a 9 by 26 game). In the examples worked, the accuracy of the approximation, measured by the approach of $\overline{V}_n$ and $\underline{V}_n$ to V related to the range of the matrix elements, seems to go essentially like $1/n$. If this is indeed so, it is extremely important for the solution of large matrices, by virtue of the fact that each iterative step takes a number of operations proportional to the linear dimension of the matrix, rather than to a higher power of the dimension, as in other computation schemes which have been suggested. Granting that for very high accuracies convergence like $1/n$ becomes painful, it may be possible to use this method to get near, and some other method to finish, the calculation. It should also be pointed out that extreme precision is ordinarily not required in practical applications.

The calculations required by this method are extremely simple. Calculation of optimum strategies at each step does not require normalization of the mixed strategy and payoff to make the sum of the probabilities unity. Instead, we may simply cumulate the vector payoff against i_n into the last cumulative vector sum, and similarly for the payoffs against j_n. Division of the maxima and minima by n to get $\overline{V}_n$ and $\underline{V}_n$ need not always be performed, but in any case this operation would be negligible in time compared with the vector addition if the dimensions of the game matrix were large.

The attached calculation consists of 25 steps carried out for the 4 by 3 matrix in Table I and illustrates typical behavior of this iteration. The values of $\underline{V}_n$ and $\overline{V}_n$ were calculated for each step to show the progress of the calculation in the early stages. The initial choice of $i_1 = 2$ is an

[1] It has since been shown by Julia Robinson that iterations of this kind must converge. Her result will appear in *The Annals of Mathematics*.

unfavorable choice with respect to minimum payoff to the first player for a single strategy. In case of ties the lowest index has been chosen.

TABLE I. GAME MATRIX

$i \backslash j$	1	2	3
1	3	1.1	1.2
2	1.3	2	0
3	0	1	3.1
4	2	1.5	1.1

Note particularly that, while $\overline{V}_n$ and $\underline{V}_n$ jog around considerably, the difference between the maximum element on one side and the minimum element on the other $(n\overline{V}_n - n\underline{V}_n)$ stays practically constant.

TABLE II. CUMULATIVE PAYOFFS

n	i_n	$j = 1$	$j = 2$	$j = 3$	$\underline{V}_n$	$\overline{V}_n$	j_n	$i = 1$	$i = 2$	$i = 3$	$i = 4$
1	2	1.3	2.0	*0*	0	3.1	3	1.2	0	*3.1*	1.1
2	3	*1.3*	3.0	3.1	0.65	2.1	1	*4.2*	1.3	3.1	3.1
3	1	4.3	*4.1*	4.3	1.37	1.77	2	*5.3*	3.3	4.1	4.6
4	1	7.3	*5.2*	5.5	1.30	1.60	2	*6.4*	5.3	5.1	6.1
5	1	10.3	*6.3*	6.7	1.26	1.52	2	7.5	7.3	6.1	*7.6*
6	4	12.3	*7.8*	7.8	1.30	1.55	2	8.6	*9.3*	7.1	9.1
7	2	13.6	9.8	*7.8*	1.11	1.46	3	9.8	9.3	*10.2*	10.2
8	3	13.6	*10.8*	10.9	1.35	1.46	2	10.9	11.3	11.2	*11.7*
9	4	15.6	12.3	*12.0*	1.33	1.59	3	12.1	11.3	*14.3*	12.8
10	3	15.6	*13.3*	15.1	1.33	1.53	2	13.2	13.3	*15.3*	14.3
11	3	15.6	*14.3*	18.2	1.30	1.48	2	14.3	15.3	*16.3*	15.8
12	3	15.6	*15.3*	21.3	1.28	1.44	2	15.4	*17.3*	17.3	17.3
13	2	*16.9*	17.3	21.3	1.30	1.48	1	18.4	18.6	17.3	*19.3*
14	4	18.9	*18.8*	22.4	1.34	1.49	2	19.5	20.6	18.3	*20.8*
15	4	20.9	*20.3*	23.5	1.35	1.51	2	20.6	*22.6*	19.3	22.3
16	2	*22.2*	22.3	23.5	1.39	1.52	1	23.6	23.9	19.3	*24.3*
17	4	24.2	*23.8*	24.6	1.40	1.52	2	24.7	*25.9*	20.3	25.8
18	2	25.5	25.8	*24.6*	1.37	1.49	3	25.9	25.9	23.4	*26.9*
19	4	27.5	27.3	*25.7*	1.35	1.47	3	27.1	25.9	26.5	*28.0*
20	4	29.5	28.8	*26.8*	1.34	1.48	3	28.3	25.9	*29.6*	29.1
21	3	*29.5*	29.8	29.9	1.40	1.49	1	*31.3*	27.2	29.6	31.1
22	1	32.5	*30.9*	31.1	1.40	1.48	2	32.4	29.2	30.6	*32.6*
23	4	34.5	32.4	*32.2*	1.40	1.47	3	33.6	29.2	*33.7*	33.7
24	3	34.5	*33.4*	35.3	1.39	1.47	2	34.7	31.2	34.7	*35.2*
25	4	36.5	*34.9*	36.4	1.40	1.47	2	35.8	33.2	35.7	*36.7*