J. Indian Inst. Sci., Jan.–Feb. 1997, 76, 47–61.

Solution of Goore game using modules of stochastic learning automata

M. A. L. THATHACHAR* AND M. T. ARVIND'
Dept. of Electrical Engg., Indian Institute of Science, Bangalore 560 012, India.
*e-mail :malt@vidyut.ee.iisc.ernet.in, 'e-mail :empty@vidyut.ee.iisc.ernet.in

Abstract

The Goore game among learning automata serves as a model for collective decision making under uncertainty. It can also be used as a tool for stochastic optimization of a function of a discrete variable. This paper presents the analysis of the Goore game where each player uses the L_{R-I} algorithm. A one-to-one correspondence is established between stable stationary points of the algorithm and the Nash equilibria of the game. A parallel algorithm involving a module of learning automata for each player is then presented with the objective of improving the speed performance. A brief analysis of the algorithm is followed by simulation studies that demonstrate the efficacy of the parallel approach.

Keywords: Learning automata, cooperative games, Goore game, stochastic optimization, parallel learning algorithm.

1. Introduction

The Goore game, formulated by Tsetlin[1], is a simple symmetric game played by several players. It is a special form of a cooperative game in which all players have identical action sets of two actions each, denoted by α_1, α_2. There are N players and at every instant k, each player i, $i = 1, 2,..., N$ chooses his action $\alpha_i(k) \in \{\alpha_1, \alpha_2\}$. An identical random payoff $\beta(k)$, with unknown distribution, is given to all the players based on the number of players selecting α_1. The players update their state depending on the payoff, and the procedure repeats itself at $k + 1$. The objective is to asymptotically choose the right number of choices of α_1 such that $\beta(k)$ is maximized in the expected sense.

Although a very specific form of a cooperative game, the Goore game can be considered a simplistic model of several real-life situations. Consider the example of recruitment of motor units[1], say in a working muscle, to perform a certain task, such as exerting a force to lift a weight. Each motor unit contributes either a fixed magnitude of force or none at all. Depending on the nature of the job, it will be necessary to recruit the correct number of motor units. It is not efficient to use more motor units than actually needed, as this will exert more force than necessary. On the other hand, if less than the required number are employed, they may not be able to perform the task at all. The problem is thus one of employing the right number of working units to perform the task.

Another application is the problem of discrete stochastic optimization in one variable, with bounded domain and range. The domain, without loss of generality, could be taken as a subset of [0,1], a linear map being employed to give a one-to-one correspondence between this set and the actual domain. It is assumed that the independent variable is discretised to provide sufficient accuracy, and the number of discretisations is the same as the number of players.

Solution of the Goore game using finite state learning automata (FSLA) has been considered in detail[2-4]. In particular, extensive studies of the final distributions of action probabilities for an infinitely increasing number of players, as functions of their memory capacities, are considered. It is shown that the group of FSLA possesses the property of asymptotic optimality.

This paper presents the solution to the Goore game using variable structure learning automata (VSLA)[5,6]. In contrast to FSLA that employ a large number of states, VSLA just maintain an action probability distribution (which is used to select actions at every instant) as their internal state, and refines this based on the action-payoff combination. The refinements are weighted by a real number b, known as the learning parameter. In most of the learning algorithms it is necessary that $b \in (0,1)$. This paper focuses attention on the analysis of the Goore game, with the players using the $L_{R\text{-}I}$ learning algorithm[5,6]. Henceforth, VSLA will be called LA for simplicity.

A common payoff game among LA will have each player represented by an LA with its actions corresponding to the player's strategies. Schemes wherein each LA uses the $L_{R\text{-}I}$ algorithm have been studied[7,8]. It has been demonstrated[7] that whenever the expected payoff matrix is unimodal, this scheme is ε-optimal. In case of multimodal payoff matrices, it is shown that all stable stationary points of the algorithm are Nash equilibria of the game[8].

The Goore game clearly falls under the multimodal category; all those player combinations resulting in the optimum number of players are stable equilibrium points for the associated ODE. This fact is later brought out in the section on analysis. However, because of the special structure of the game, it is possible to derive a tighter relationship between the stable stationary points of the algorithm and the Nash equilibria of the game. In fact, in the section on analysis, it is shown that they have a one-to-one correspondence.

Whenever a high degree of accuracy is required in a Goore game (a typical requirement in the context of an optimization set up), it becomes necessary to increase the number of players. Even with a reasonable increase in the number of players, the value of the learning parameter needed for good accuracy is drastically reduced, adversely affecting the speed of convergence of the algorithm. To improve the speed performance without sacrificing the accuracy of the learning procedure, an algorithm for operating a group of LA in parallel was proposed[9]. An extension of this algorithm is proposed in this paper for solving the Goore game. The scheme will be analysed and its efficacy demonstrated by means of extensive simulation studies.

The paper is organised as follows. Section 2 formulates the Goore game problem and presents the $L_{R\text{-}I}$ algorithm for solving it. The algorithm is analysed in Section 3 by using weak convergence theory[10]. Section 4 presents the parallel algorithm for Goore game and its analysis. Simulation studies for both the schemes are presented in Section 5. An extension of the Goore game to multiple groups of players is proposed in Section 6. An analysis with L_{R-I} algorithm is carried out; a partial characterization of the equilibrium points of the ODE is provided. Section 7 concludes the paper.

2. Problem formulation and algorithm

The Goore game among N players, with each player employing the L_{R-I} algorithm, to choose among his strategies, is analysed in this section. In the simplest form, player i, $i = 1, 2 \ldots, N$ chooses his action $\alpha_i(k)$ from two actions α_1, α_2 available to him,

$$i.e., \alpha_i(k) \in \{\alpha_1, \alpha_2\};\ i = 1, 2 \ldots, N.$$

The LA approach to solve this game comprises LA representing each player, each equipped with an action probability for action selection. Let $x_i(k)$ be the probability that player i selects first action α_1 at instant k. Correspondingly, probability of player i selecting second action is $(1 - x_i(k))$. The action probability vector and the action vector, respectively, are

$$\mathrm{x}(k) \triangleq [x_1(k), x_2(k), \ldots, x_N(k)];$$
$$\alpha(\mathrm{k}) \triangleq [\alpha_1(k), \alpha_2(k), \ldots, \alpha_N(k)].$$

Let $n_1(k)$ be the total number of times the first action is selected at k.

$$i.e., n_1(k) \triangleq \sum_{j=1}^{N} \mathrm{I}\{\alpha_j(k) = \alpha_1\}$$

where $\mathrm{I}\{A\}$ is the indicator function of event A.

The environment gives a reinforcement signal $\beta(k) \in [0,1]$ based on $n_1(k)$. The goal of the players is to maximize $E[\beta(k)]$ by choosing the appropriate number of first actions collectively. In the most general situation, a player might not even be aware of the existence of other players or the number of players involved. Each player need know only the payoff, once he chooses his action.

It is assumed that each player makes use of the L_{R-I} algorithm[5].

$$x_i(k+1) \triangleq \begin{cases} x_i(k) + b\beta(k)(1 - x_i(k)) & \text{if } \alpha_i(k) = \alpha_1; \\ x_i(k) - b\beta(k)x_i(k) & \text{else.} \end{cases} \qquad (1)$$

The following notations are convenient during the analysis of the above algorithm. In the sequel, explicit dependence on k is not shown whenever there is no scope for ambiguity.

$$\Delta x_i(k) \triangleq E[x_i(k+1) - x_i(k) | \mathrm{x}(k)];$$
$$\Delta \mathrm{x} \triangleq [\Delta x_1, \Delta x_2, \ldots, \Delta x_N].$$

The following assumptions are made:

Assumption 1. *The reinforcement signal $\beta(k)$ is non-negative and bounded. Without loss of generality,* $\beta \in [0,1]$. Otherwise if $\beta \in [0,M]$ for some $1 < M < \infty$, β in the algorithm is replaced by β/M.

Assumption 2. *The expected payoff* $g : [0,1] \mapsto [0,1]$ *is continuous. This is necessary as* $g(\cdot)$ *should not have jumps for any given number of players.*

Assumption 3. $g(i/N)$, $i = 0,1,2,\ldots, N$, *is unimodal. The case of multimodal* $g(\cdot)$ *is commented upon later.*

Here,

$$E[\beta | n_1] = g\left(\tfrac{n_1}{N}\right).$$

In the sequel, g_i denotes $g(i/N)$ for convenience.

Remark 1. *Given N, the problem essentially reduces to that of finding the maximizer of* $g\left(\frac{i}{N}\right)$; $i = 0,1,\ldots, N$. *Hence greater the N, finer the approximation to the actual maximizing value.*

Remark 2. *Since the environment payoff depends on the number of players choosing the first action, and not their identity, all combinations that give the appropriate number* n_1 *maximize* $g(\cdot)$.

Remark 3. *By virtue of Assumption* 3, $\exists\ l$ *s.t* $g_l > g_i$, $\forall i \neq l$ *and one of the following is true.*

- $l = 0$ *with* $g_0 > g_1 > \ldots > g_N$
- $l = N$ *with* $g_0 < g_1 < \ldots < g_N$
- $g_{j-1} < g_j\ \forall\ 0 < j \leq l$ *and* $g_j < g_{j-1}\ \forall\ l < j \leq N$.

It is easy to see that combinations corresponding to l choices of α_1 *are the only Nash equilibria for the game.*

3. Analysis

Definitions

$$S(N) \triangleq \{1,2,\ldots, N\}$$

$$S(N,i) \triangleq \{1,2,\ldots, i-1, i+1,\ldots, N\}$$

$$\mathcal{P}(N) \triangleq \{W : W \subseteq S(N)\}$$

$$\mathcal{P}(N,i) \triangleq \{W : W \subseteq S(N,i)\}$$

The probability that α_1 is chosen l times is given by

$$\Pr\{n_1(k) = l | x, \alpha_i(k) = \alpha_1\} = \sum_{W \in \mathcal{P}(n,i); |W| = l-1} \left(\prod_{m \in W} x_m \prod_{n \in S(N,i)-W} (1 - x_n) \right) \tag{2}$$

$$\Pr\{n_1(k) = l|x, \alpha_i(k) = \alpha_2\} = \sum_{W\in\mathcal{P}(N,i):|W|=l}\left(\prod_{m\in W} x_m \prod_{n\in S(N,i)-W}(1-x_n)\right) \tag{3}$$

for player i choosing his first and second actions respectively. From the algorithm,

$$\Delta x_i = bx_i\,(1-x_i)E[\beta|\alpha_i = \alpha_1,\mathbf{x}] - bx_i(1-x_i)\,E[\beta|\alpha_i = \alpha_2,\mathbf{x}] = bx_i(1-x_i)f_i(\mathbf{x}) \tag{4}$$

where

$$f_i(\mathbf{x}) = \sum_{l=0}^{n-1}(g_{l+1} - g_l)\left(\sum_{W\in\mathcal{P}(N,i):|W|=l}\left(\prod_{m\in W} x_m \prod_{n\in S(N,i)-W}(1-x_n)\right)\right) \tag{5}$$

with

$$\prod_{\phi} \triangleq 1.$$

Example: $N = 3$, $S(3,1) = \{2,3\}$

$$f_1(\mathbf{x}) = (g_1 - g_0)(1-x_2)(1-x_3) + (g_2 - g_1)\,(x_2(1-x_3) + x_3(1-x_2)) + (g_3 - g_2)x_2\,x_3.$$

Remark 4. *$f_i(\cdot)$ is not a function of x_i. $f_i(\cdot)$ are symmetric, in the sense that, $f_i(\cdot)$ is obtained by substituting x_j* for x_i in $f_j(\cdot)$.

Now, $\forall b > 0$, $\{\mathbf{x}(k):k > 0\}$ is a Markov process with dynamics dependent on b. This dependence is explicitly denoted by $\mathbf{x}^b(k)$. Define continuous time interpolations $\mathbf{X}^b(t)$ of $\mathbf{x}^b(k)$ by

$$\mathbf{X}^b(t) = \mathbf{x}^b(k) \text{ if } t \in [kb,(k+1)b).$$

The algorithm can be written as

$$\mathbf{x}^b(k+1) = \mathbf{x}^b(k) + bG(\mathbf{x}^b(k),\ \theta^b(k))$$

where θ comprises the action vector and the payoff.

The following conditions are satisfied by the algorithm:

1. $\{\mathbf{x}(k),\ \theta(k-1) : k \geq 0\}$ is a Markov process.
2. The outputs of the automata are from finite sets. The payoff takes values from the closed interval $[0,1]$. Thus $\theta(k)$ takes values from a compact metric space S.
3. The function $G(\cdot,\cdot)$ is bounded, continuous and independent of b.
4. Let B be a Borel subset of S (defined above). The one-step transition function $TF(\cdot)$, defined as

$$TF(\theta,1,B|\mathbf{x}) \triangleq \text{Prob}\{\theta(k) \in B|\theta(k-1) = \theta,\ \mathbf{x}(k) = \mathbf{x}\}$$

is independent of b, k and θ. Thus

$$TF(\theta,1,B|\mathbf{x}) = TF(B|\mathbf{x}).$$

5. $TF(\theta,1,\cdot|\mathbf{x})$ is its own unique invariant probability measure, since it is independent of θ. Denote this invariant probability measure by $M(\mathbf{x})$. As S is compact, the set of probability measures $M(\mathbf{x})$ is trivially tight.
6. $\int G(\mathbf{x},\theta') \, TF(\theta,1,d\theta'|\mathbf{x})$ is independent of θ and continuous with respect to $\mathbf{x}$.

With the above conditions satisfied the following theorem results from weak convergence theory[10].

Theorem 1. *For the L_{R-I} algorithm, sequence of interpolated processes* $\{\mathbf{X}^b(\cdot) : b > 0\}$ *converges weakly as $b \to 0$ to* $\mathbf{z}(\cdot)$, *given by*

$$\frac{d\mathbf{z}}{dt} = \left[h_1(\mathbf{z}), h_2(\mathbf{z}), \ldots, h_N(\mathbf{z})\right]; \mathbf{z}(0) = \mathbf{x}(0)$$

$$\text{where } h_i(\mathbf{z}) = z_i(1 - z_i) f_i(\mathbf{z});\ i = 1,2,\ldots, N. \tag{6}$$

3.1. *Equilibrium points of the ODE*

The equilibrium points of the system (6) are obtained by setting $\frac{d\mathbf{z}}{dt} = 0$. The solutions can be categorised as follows:

- All $\mathbf{z} \in \{0,1\}^N$.
- All $z_0 \in (0,1)$ with $\mathbf{z} = [z_0, z_0, \ldots, z_0]$ and $f_i(\mathbf{z}) = 0 \forall i$.
- Combinations of the above categories; *i.e.*, $z_i \in \{0,1\}$ for some of the is and $z_i = z_0 \in (0,1)$ for the rest.

From the expression for h_i, we have

$$\begin{aligned} \frac{\partial h_i}{\partial z_j} &= z_i(1-z_i)\frac{\partial f_i}{\partial z_j}; \forall j \neq i \\ \frac{\partial h_i}{\partial z_i} &= (1-2z_i)f_i \end{aligned} \tag{7}$$

To examine the local stability of each equilibrium point, the eigenvalues of the matrix $\left[\frac{\partial h_i}{\partial z_j}\right]$; $1 \le i, j \le N$, evaluated at each of these points, are considered.

Case 1: *N bit binary strings*

In this case $z_j \in \{0, 1\}$; $\forall j$. Hence

$$\begin{aligned} \frac{\partial h_i}{\partial z_j} &= 0; \forall j \neq i \\ \frac{\partial h_i}{\partial z_i} &= \begin{cases} f_i(\mathbf{z}) & \text{if } z_i = 0 \\ -f_i(\mathbf{z}) & \text{if } z_i = 1 \end{cases} \end{aligned} \tag{8}$$

Consider a solution string with l 1s and $z_i = 1$. For this combination it is easy to check from (5) that,

$$\frac{\partial h_i}{\partial z_i} = -(g_l - g_{l-1}).$$

Correspondingly, if there are l 1s with $z_i = 0$,

$$\frac{\partial h_i}{\partial z_i} = (g_{l+1} - g_l).$$

Hence for such a string $\left[\frac{\partial h_i}{\partial z_i}\right]$ is a diagonal matrix with $-(g_l - g_{l-1})$ for l diagonal entries and $(g_{l+1} - g_l)$ for $(n - l)$ diagonal entries. Therefore, only for that l given by Remark 3, both the eigenvalues are negative, and subsequently, only those solutions that contain l 1s are asymptotically stable.

Case 2: $z_0 \in (0,1)$ *with* $f_i(\mathbf{z}) = 0$; $\forall_i$; $\mathbf{z} = [z_0, z_0, \ldots, z_0]$

In this case,

$$\begin{aligned} &\frac{\partial h_i}{\partial z_i} = 0;\ \text{for each } z_0 \\ &\frac{\partial h_i}{\partial z_j} = z_0(1 - z_0)\frac{\partial f_i}{\partial z_j} = a(z_0); a(z_0) \in \mathcal{R}. \end{aligned} \tag{9}$$

Consequently the matrix $\left[\frac{\partial h_i}{\partial z_j}\right]$ has 0s on the diagonal, and all other entries as $a(z_0)$. The matrix is symmetric and all its eigenvalues are real. In fact, the eigenvalues of such a matrix are $(N - 1)\ a(z_0)$ and $-a(z_0)$. Hence, irrespective of the sign of $a(z_0)$, one of the eigenvalues is always positive, and the corresponding equilibrium point is unstable. The same holds $\forall z_0 \in (0,1)$ *s.t.* $f_i(z_j = z_0;\ \forall j) = 0;\ \forall i$. Alternatively, since the sum of the diagonal entries which is the sum of all the eigenvalues, is 0, and the matrix itself is symmetric, some eigenvalue must be positive real and hence the instability.

Case 3:$z_j \in \{0,1\}$ *for utmost* $(N - 2)$ *of the js and* $z_j = z_0$, s.t. $f_j = 0$ *for the rest, with* $z_0 \in (0,1)$

In this case, the matrix $\left[\frac{\partial h_i}{\partial z_j}\right]$ has the form $\begin{bmatrix} D & 0 \\ B & C \end{bmatrix}$ where D is diagonal and corresponds to z_js that are binary; and C is a symmetric matrix with zeros on the principal diagonal and corresponds to all those $z_j \in (0,1)$. The eigenvalues of such a matrix are the eigenvalues of D along with those of C. Similar arguments as in Case 2 lead to some of the eigenvalues of C having positive real parts and hence this case does not result in any stable equilibrium points.

Remark 5. From the foregoing analysis it is possible to draw a few conclusions when $g(\cdot)$ is multimodal. Consider the set

$$U \underline{\underline{\Delta}}\ \{i: g_{i-1} < g_i \text{ and } g_{i+1} < g_i;\ 0 < i < N\}.$$

Include 0 in U if $g_0 > g_1$. Also include N in U if $g_N > g_{N-1}$. It can be verified that the only Nash equilibria of the game are all those combinations that result in i choices of α_1, $\forall i \in U$. It is easy to see that all those string combinations containing 1 in i positions, $i \in U$, will have all the corresponding eigenvalues negative, and hence, each of these combinations is stable. This establishes the equivalence between the Nash equilibria of the game and the stable equilibrium points of the ODE for the multimodal case.

4. Parallel algorithm

As the number of players in the Goore game increases, the value of the learning parameter b needs to be drastically reduced if good accuracy (in terms of less number of wrong convergences) is to be maintained. But reduction in the value of the learning parameter will adversely affect the speed of convergence of the algorithm. To maintain good accuracy levels while increasing the speed of convergence, an algorithm for the parallel operation of several LA was proposed[9]. This scheme could be regarded as an extension of the L_{R-I} algorithm, as the algorithm is the same as L_{R-I} when only one LA is present. The improvement in the speed of convergence due to the parallel operation is theoretically established and demonstrated by simulation studies[9].

The game version of the above algorithm is proposed in this section to solve the Goore game with improved speed performance. The scheme is described below and an outline of the analysis is presented. Justification is also provided for the improved speed performance.

4.1. *Algorithm A*1

In this scheme, each player i, $i = 1, 2, \ldots, N$; is replaced by n identical LA, each of which chooses actions independent of others. Such an arrangement of LA could be regarded as forming a module, each module corresponding to a player. The action probability $x_i(k)$ is common to all members of the module. The action of jth element of ith (player) module, $\alpha_i^j(k) \in \{\alpha_1, \alpha_2\}$.

$$\Pr\{\alpha_i^j(k) = \alpha_1\} = x_i(k) = 1 - \Pr\{\alpha_i^j(k) = \alpha_2\}; 1 \le j \le n, 1 \le i \le N.$$

There are n simultaneous plays of the Goore game by n teams of LA, each team being formed by the jth member of each module ($j = 1, 2, \ldots, n$). The payoff to the action combination $\alpha^j(k) = [\alpha_1^j(k), \alpha_2^j(k), \ldots, \alpha_N^i(k)]$ is denoted by $\beta^j(k)$. $\beta^j(k) \in [0,1]$, and it depends on $n_1^j(k)$, the number of times α_1 is selected by jth members of all modules. Other notations used in the algorithm are summarised below.

- Total reward to α_1 for player i at k : $q^i(k) \underline{\underline{\Delta}} \sum_{j=1}^n \beta^j(k) I\{\alpha_i^j(k) = \alpha_1\}$.
- Total reward to each player at k : $q(k) \underline{\underline{\Delta}} \sum_{j=1}^n \beta^j(k)$.
- Normalised learning parameter: $\tilde{b} \underline{\underline{\Delta}} \frac{b}{n}$.

*Algorithm A*1

The algorithm to update the action probabilities is

$$x_i(k+1) = x_i(k) + \tilde{b}\left(q^i(k) - q(k)x_i(k)\right); i = 1,2,\ldots,N. \tag{10}$$

For each player, the algorithm compares the fraction of the total payoff obtained by α_1 to its probability of selection and increases (decreases) the latter if the former is larger (smaller).

4.2. *Analysis*

Results similar to those derived for L_{R-I} algorithm are obtained in this subsection for Algorithm A1. Taking expectations on both sides of Algorithm A1 conditioned on $x(k)$

$$\Delta x_i(k) = \frac{b}{n} E\left[q^i(k) - x_i(k)q(k) | \mathbf{x}(k)\right]. \tag{11}$$

Substituting for q^i and q in (11),

$$\Delta x_i = \frac{b}{n} \sum_{j=1}^{n} E\left[\beta^j\left(I\{\alpha_i^j = \alpha_1\} - x_i\right) | \mathbf{x}\right]. \tag{12}$$

Remark 6. $\beta^j(k)$ does not depend on $\alpha^s(k)$; $s \neq j$.

Now,

$$E\left[\beta^j\left(1\{\alpha_i^j = \alpha_1\} - x_i\right) | \mathbf{x}\right] = E\left[\beta^j(1 - x_i) | \mathbf{x}, \alpha_i^j = \alpha_1\right]x_i - (1 - x_i)E\left[\beta^j x_i) | \mathbf{x}, \alpha_i^j = \alpha_2\right]$$

$$= x_i(1 - x_i)f_i(\mathbf{x}) \tag{13}$$

The last equality follows because of independent selection of actions, and $f_i(\mathbf{x})$ is as given by (5). Thus

$$\Delta x_i = \tilde{b} n x_i(1 - x_i)f_i(\mathbf{x}). \tag{14}$$

Following arguments similar to those of the previous section (this also involves checking the conditions therein), the following theorem results:

Theorem 2. *For the Algorithm A*1, *sequence of interpolated processes* $\left\{\mathbf{X}^{\tilde{b}}(\cdot) : \tilde{b} > 0\right\}$ *converges weakly as* $\tilde{b} \to 0$ *to* $\mathbf{z}(\cdot)$, *given by*

$$\frac{d\mathbf{z}}{dt} = \left[h_1(\mathbf{z}), h_2(\mathbf{z}), \ldots, h_N(\mathbf{z})\right]; \mathbf{z}(0) = \mathbf{x}(0)$$

$$\text{where } h_i(\mathbf{z}) = nz_i(1 - z_i)f_i(\mathbf{z}); i = 1,2,\ldots,N. \tag{15}$$

It is easy to see that ODE (6) and ODE (15) have the same set of equilibrium points and exactly the same stability arguements hold for any given *n*.

Remark 7. *The long time behaviour of Algorithm A1 can be approximated by ODE (15). With $\tilde{b}$ same as b of L_{R-I} algorithm, the approximation is valid to a similar degree of accuracy. Within this accuracy level Algorithm A1 is faster than L_{R-I} algorithm as ODE (15) has a larger speed of convergence.*

Extensive simulations demonstrate improvements in speed of convergence for various values of *n*, and are presented in the following section.

5. Simulation studies

Simulation studies for unimodal and multimodal functions, using Algorithms L_{R-I} and A1 are reported in this section. The studies indicate good speedups for various module sizes. The studies are tabulated and discussed for the two cases. In the tabulations $n > 1$ indicates study using Algorithm A1.

5.1. *Unimodal function*

The following unimodal mean payoff function was employed for simulation studies. In this case as well as the following case the random payoff is obtained by adding a zero mean random noise arising from a uniform distribution. The uniform distributions spread between $-g(x)$ and $+g(x)$ if $g(x) \leq 0.5$; between $-(1 - g(x))$ and $+(1 - g(x))$ otherwise.

$$g(x) = 0.9\exp\left(-\frac{(x-0.3)^2}{0.01}\right); \forall x \in [0,1].$$

Based on the value of *N*, the problem is that of finding *l* for which $g(l/N)$ is maximum. For example, for $N = 4$ the mean payoffs have the value $g(\mathrm{i}/N)$; $n = 0,1,2,3,4$. It is desirable in this case that $n_1(k)$ converges to $n_1^* = 1$ as $g(1/4)$ is the maximum of $g(i/N)$; $i = 0,1,2,3,4$.

Twenty runs of simulation were performed for each *n*, *N* and *b* and Table I lists the values of the average number of iterations for convergence in each case. The value of *b* listed in each case is the maximum for which no wrong convergence resulted in any of the runs. Convergence was assumed when all x_is went outside the interval [0.01, 0.95]. The value of $x_i(0) = 0.5$ was used $\forall i = 1,2,\ldots, N$.

Table I
Unimodal function

	$N = 4$,	$n_1^* = 1$	$N = 8$	$n_1^* = 2$	$N = 16$,	$n_1^* = 5$
n	*b*	*Avg. Iter.*	*b*	*Avg. Iter.*	*b*	*Avg. Iter.*
1	0.2	84	0.01	12025	0.005	61279
2	0.5	28	0.02	6651	0.01	29893
4	1.0	13	0.04	3151	0.02	14058
8	1.0	18	0.1	1356	0.04	7331
16	1.0	21	0.16	813	0.08	3503
32	1.0	21	0.3	459	0.16	1713

Table II
Multimodal function

	$N = 4,$	$n_1^* = 0,4$	$N = 8$	$n_1^* = 0,8$	$N = 16,$	$n_1^* = 0,16$
n	b	*Avg. Iter.*	b	*Avg. Iter.*	b	*Avg. Iter.*
1	0.4	23	0.1	188	0.05	808
2	0.8	12	0.2	99	0.1	423
4	1.0	9	0.3	65	0.2	213
8	1.0	9	1.0	20	0.4	112
16	1.0	9	1.0	21	0.8	57
32	1.0	9	1.0	21	1.0	46

5.2. *Multimodal function*

The multimodal mean payoff function considered for studies is

$$g(x) = 2(x - 0.3)^2; \ \forall x \in [0,1].$$

It is obvious that $n_1^* = 0$ and N, are the optimal combinations in their neighbourhood. Wrong convergence is said to occur whenever $n_1(k)$ does not go to 0 or N. Other conditions for simulation remain the same. Simulation studies are presented in Table II for this case.

The tables demonstrate the efficacy of the parallel Algorithm A1 in terms of good speedups over the L_{R-I} algorithm. It is seen that the speedup is of the order of the module size in almost all the cases. The implications of this factor in a real life situation are quite significant; even with noisy inputs, fast convergence is achievable with good accuracies.

6. Groups of players

An extension of the Goore game to involve groups of players, is proposed in this section. The obvious application to multivariable stochastic optimization serves as a good motivation. Results obtained for single variable optimization problems encourage the investigation of the feasibility of the Goore game approach for solving multidimensional optimization problems. Only the analysis for L_{R-I} algorithm is considered; extension to the corresponding parallel case is straightforward. At present, only a partial characterization of the equilibrium points is available. The intent of this section is to highlight the difficulties involved in the analysis of the multiple group situation.

The notations used in this section sometimes do not differ from those of the section on parallel algorithm. This is done only to simplify the notation by avoiding too many subscripts and superscripts.

6.1. *Problem formulation*

Goore game among M groups of players is considered. In group j, player i chooses action $\alpha_i^j(k) \in \{\alpha_1^j, \alpha_2^j\}$. There are N_j players in group j, of which $n_j(k)$ select α_1^j at instant k.

$x_i^j(k)$ denotes the probability of the event "player i of group j selects α_1^j at instant k". The payoff depends on the fraction of players choosing the first action in each group. In the following, the dependence on k is omitted for notational convenience.

Let

$$\mathbf{x} \underline{\underline{\Delta}} \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M\right]$$

where

$$\mathbf{x}^j \underline{\underline{\Delta}} \left[x_1^j, x_2^j, \ldots, x_{N_i}^i\right].$$

Each player uses the L_{R-I} algorithm.

$$x_i^j(k+1) \underline{\underline{\Delta}} \begin{cases} x_i^j(k) + b\beta(k)\left(1 - x_i^j(k)\right) & \text{if } \alpha_i^j(k) = \alpha_1^j \\ x_i^j(k) - b\beta(k)x_i^j(k) & \text{else.} \end{cases} \tag{16}$$

Similar assumptions on $g(\cdot)$ hold, with

$$E\left[\beta \mid n_1, \ldots, n_M\right] = g\left(\frac{n_1}{N_1}, \ldots, \frac{n_M}{N_M}\right)$$

$g(\cdot)$ is denoted as $g_{n_1 n_2 \ldots n_M}$ for convenience.

Define the functions

$$q_i^j(l) \underline{\underline{\Delta}} \sum_{W \in \mathcal{P}(N_j, i); |W| = l} \left(\prod_{m \in W} x_m^j \prod_{n \in S(N_j, i) - W} \left(1 - x_n^j\right) \right);$$

$$u^j(l) \underline{\underline{\Delta}} \sum_{W \in \mathcal{P}(N_j); |W| = l} \left(\prod_{m \in W} x_m^j \prod_{n \in S(N_j) - W} \left(1 - x_n^j\right) \right);$$

Then

$$\Pr\left\{n_1 = l_1, \ldots, n_M = l_M \mid \mathbf{x}, \alpha_i^j = \alpha_1^j\right\} = q_1^j\left(l_j - 1\right) \prod_{s=1; s \neq j}^{M} u^s(l_s);$$

$$\Pr\left\{n_1 = l_1, \ldots, n_M = l_M \mid \mathbf{x}, \alpha_i^j = \alpha_2^j\right\} = q_1^j\left(l_j\right) \prod_{s=1; s \neq j}^{M} u^s(l_s);$$

From the algorithm,

$$\Delta x_i^j = b x_i^j \left(1 - x_i^j\right) f_i^j(\mathbf{x})$$

where

$$f_i^j(\cdot) \underline{\underline{\Delta}} \sum_{l_1=0}^{N_1} \cdots \sum_{l_{j-1}=0}^{N_{j-1}} \sum_{l_j=0}^{N_j - 1} \sum_{l_{j+1}=0}^{N_{j+1}} \cdots \sum_{l_M=0}^{N_M} \left(g_{l_1 \ldots l_{j-1}(l_j+1) l_{j+1} \ldots l_M}\, g_{l_1 \ldots l_M}\right) q_i^j\left(l_j\right) \prod_{s=1; s \neq j}^{M} u^s(l_s).$$

Define

$$h_i^j(\mathbf{z}) \triangleq z_i^j\left(1 - z_i^j\right) f_i^j(\mathbf{z})$$

and

$$h^j \triangleq \left[h_1^j, h_2^j, \ldots, h_{N_j}^j\right].$$

Then, as earlier,

$$\frac{d\mathbf{z}}{dt} = \left[h^1, h^2, \ldots, h^M\right]$$

is the associated ODE of the system. The results pertaining to the weak convergence of the interpolated processes to the associated ODE can be obtained here also.

From the expression for h_i^j,

$$\frac{\partial h_i^j}{\partial z_i^j} = \left(1 - 2z_i^j\right) f_i^j.$$

Similarly, $\frac{\partial h_i^j}{\partial z_n^j}; n \neq i$ is obtained by replacing q_i^j by $\frac{\partial q_i^j}{\partial z_n^j}$ in the expression for h_i^j and $\frac{\partial h_i^j}{\partial z_n^m}$; $m \neq j$ is obtained by replacing h^m by $\frac{\partial h^m}{\partial z_n^m}$ in the expression for h_i^j.

6.2. *Equilibrium points of the ODE*

Denote

$$A \triangleq \left[\frac{\partial h_i^j}{\partial z_n^m}\right].$$

The following solutions of the ODE are characterized in this subsection. For every $J \in \mathcal{P}(M)$,

$$z_i^j = a \in (0,1); s.t.\, f_i^j = 0, \forall i = 1,2,\ldots,N_j; j \in J \text{ and } \mathbf{z}^j \in \{0,1\}^{N_j}; j \in S(M) - J.$$

The following cases arise while observing the local stability of the equilibrium points.

Case 1: $J = \phi$

$j \in S(M)$; $\forall j$ and hence A is diagonal. The analysis similar to Case 1 of the single group situation holds, and only those solution combinations corresponding to the single maximum (of the discretised function) are stable.

Case 2: $J = S(M)$

$z_i^j = a^j \in (0,1); \forall i, j$. From the expression for the derivatives,

$$\frac{\partial h_i^j}{\partial z_i^j} = 0$$

Hence, A is a symmetric matrix with zero entries on the principal diagonal. Since sum of the eigenvalues of such a matrix is zero and the matrix is nonsingular, some of the eigenvalues are positive real. Hence these combinations are not stable.

Case 3: $J \neq \phi$; $J \neq S(M)$

Whenever $\mathbf{z}^j \in \{0,1\}^{N_j}$, from the above derivative expressions $\frac{\partial h_i^j}{\partial z_n^j} = 0; \forall n \neq i$, and $\frac{\partial h_i^j}{\partial z_n^m} = 0; \forall m \neq j$. Suitable permutations can be carried out on the A matrix to bring it to the form $\begin{bmatrix} D & 0 \\ B & C \end{bmatrix}$ where D is diagonal and corresponds to $j \in S(M)-J$, and C is a symmetric matrix with zeros on the principal diagonal and corresponds to $j \in J$. The eigenvalues of such a matrix are the eigenvalues of D along with those of C. Similar arguments as in Case 2 lead to some of the eigenvalues of C having positive real parts.

Remark 8. *The equilibrium points not characterized in this section are those in which some of the values in each group are binary and the others identical and belonging to the interval* (0,1). *Obtaining the eigenvalues of the matrix of partial derivatives does not seem to be as simple as it was in the previous sections.*

Remark 9. *For the class of equilibrium points considered, more can be said in case of multimodal functions. As in the single group situation, the set U can be constructed* (*refer Remark* 5), *and all those binary combinations corresponding to the elements of this set are seen to be stable.*

7. Conclusions

A detailed analysis of the Goore game among LA has been presented. The equivalence of the stable equilibrium points of the associated ODE and the Nash equilibria of the game was demonstrated. A weak convergence result is employed to show that the long-time behaviour of the algorithm could be approximated by the associated ODE for small learning parameters. A parallel algorithm has been presented to improve the speed performance of the Goore game. Similar results regarding the stability of the equilibrium points have been derived for this algorithm. Simulation studies have been presented to demonstrate the improvements in speed performance for the parallel algorithm. Further improvements in speed appear possible by considering a larger number of teams formed by different combinations of members of the n available modules. Finally, extension of the analysis to the multiple group situation is considered, with applications to multivariable stochastic optimization in mind. A partial characterization of the equilibrium points of the associated ODE is presented. Further efforts will be directed at providing a complete characterization of the multiple group Goore game.

References

1. TSETLIN, M. L. *Automata theory and modeling of biological systems*, 1973. Academic Press.

2. Borovikov, V. A. and Bryzgalov, V. I. A simple symmetric game between many automata, *Avomat. Telemekh.*, 1965, 26(4).

3. Volkonskiy, V. A. Asymptotic properties of the behaviour of simple automata in a game, *Probl. Peredachi Inform*, 1965, 1(2).

4. Pittel, B. G. The asymptotic properties of a version of the goore game. *Probl. Peredachi Inform.*, 1965, 1(3).

5. Narendra, K. S. and Thathachar, M. A. L. *Learning automata: An introduction*, 1989, Prentice Hall.

6. Lakshmivarahan, S. *Learning algorithms: Theory and applications*, 1981, Springer Verlag.

7. Wheeler, Jr., R. M. and Narendra, K. S. Decentralized learning in finite markov chains. *IEEE Trans.*, 1986, AC-31, 519–526.

8. Sastry, P. S., Phansalkar, V. V. and Thathachar, M. A. L. Decentralized learning of nash equillibria in multi—person stochastic games with incomplete information. *IEEE Trans.*, 1994, SMC-24, 769–777.

9. Thathachar, M. A. L. and Arvind, M. T. A parallel algorithm for operating a stack of learning automata. *Proc. Fourth Intelligent Systems Symp.*, IEEE Bangalore Section, Nov 1994.

10. Kushner, H. J. *Approximation and weak convergence methods for random processes*, 1984, MIT Press.