**Les cahiers du GERAD** ISSN: 0711–2440

# Approximative Solutions to Continuous Stochastic Games

M. Breton, P. L'Écuyer

G–90–25

May 1990

# Approximative Solutions to Continuous Stochastic Games*

Michèle Breton
GERAD and École des Hautes Études Commerciales

Pierre L'Écuyer
Dept. d'Informatique et de Recherche Opérationnelle
Université de Montréal

May 1990

*Research supported by NSERC-Canada, Grants #OGPIN020 and #A5463, and FCAR-Québec Grants #90-NC-0252 and #EQ2831.

## Abstract

In this paper, we suggest an approximation procedure for the solution of two-player zero-sum stochastic games with continuous state and action spaces similar to finite elements and modified policy iteration approaches used for the solution of Markov Decision Problems.



## Résumé

Dans cet article, nous suggérons une méthode d'approximation pour la solution de jeux stochastiques à somme nulle où les espaces d'états et d'actions sont continus; l'algorithme proposé s'apparente aux méthodes d'éléments finis ainsi qu'aux procédures itératives modifiées utilisées pour la solution de processus de décision markoviens.

# 1 Introduction

Zero-sum, two-player discounted stochastic games were introduced by L. S. Shapley [13] who gave a (constructive) existence proof of saddle points in stochastic games with finite state and action sets which provided a first iterative algorithm for the computation of the value of such games. Since then, other iterative algorithms have been proposed (Pollatscheck and Avi-Itzhak [11], Filar and Tolwinsky [6], Tolwinsky [14]); all these algorithms can be related to methods used for the solution of Markov Decision Problems, i.e. value iteration, policy iteration and modified policy iteration.

We are interested in devising an approximation procedure to solve stochastic games with continuous (or very large) state or action spaces. There already exists a large literature on discretization and approximation methods in dynamic programming (see [9] and the references cited there). Approximation methods are used in order to define "smaller" problems which can then be solved using any available algorithm for discrete problems. For dynamic games, theoretical considerations pertaining to their approximation have been presented by [15]; we will address here a general method for the computation of an equilibrium point in a zero-sum game.

In this paper, we consider a two-player zero-sum stochastic game model with continuous state and action spaces as studied in [10] (the extension of the approximation procedure to more general models, e.g. locally contracting renewal games (see [3]) is straightforward). We describe a finite element computational approach to deal with continuous or very large state spaces. The algorithm used with the finite element approach can be viewed as an extension of the "modified policy iteration" algorithm studied in [12] for Markov Decision Problems.

The outline of the paper is as follows: In section 2, we state the basic stochastic game model and its associated dynamic programming operators. In section 3, we briefly recall some existing algorithms for the solution of discrete stochastic games. Finally, in section 4, we describe a finite element computational approach, using an approximate policy iteration algorithm.

# 2 Zero-sum Two-Player Stochastic Game model

Consider the following two-player game model with Borel state space $S$ and separable metric action spaces $A$ and $B$. For each state $s$ in $S$, let $A(s) \subset A$ and $B(s) \subset B$ be the non empty compact set of admissible actions to player 1 and player 2 respectively when the system is in state $s$. To allow for randomized strategies, we assume that each action in $A(s)$ and $B(s)$ is in fact a mixed action, i.e. a probability measure over an

underlying set of pure actions. At each of an infinite sequence of stages (decision times), the players observe the state $s$ of the system and independently select actions $a \in A(s)$ and $b \in B(s)$. For the current stage, the expected return to player 1, paid by player 2, is $r(s,a,b)$ and the system moves to a new state $s'$ according to a probability measure $q(\cdot|s,a,b)$ over $S$. A new action pair is then selected by the players, and so on. The expected one-stage return function of player 1 $r(s,a,b)$ is a bounded Borel-measurable real valued function of $s \in S$, $a \in A(s)$ and $b \in B(s)$ and the law of motion is given by the family of probability measures $\{q(\cdot|s,a,b) : s \in S, a \in A(s), b \in B(s)\}$ which form a Borel-measurable stochastic kernel on $S$ given $s \in S$, $a \in A(s)$ and $b \in B(s)$.

A policy $\delta$ for player 1 is a Borel-measurable function from $s \in S$ into his admissible action set $A(s)$ under which player 1 takes the mixed action $\delta(s)$ whenever the system is in state $s$. In the same way, a policy $\gamma$ for player 2 is a Borel-measurable function $\gamma : s \in S \rightarrow \gamma(s) \in B(s)$. Let $\Delta$ and $\Gamma$ denote the set of policies for player 1 and 2 respectively. A stationary strategy pair for players 1 and 2, denoted $[\delta, \gamma]$, consists in using respectively the policies $\delta$ and $\gamma$ at each stage of the game. In this paper, we consider only stationary strategies.

Let $v_{[\delta,\gamma]}(s)$ denote the expected discounted sum of the rewards of player 1 when the initial state of the system is $s$ and the players use the stationary strategy pair $[\delta, \gamma]$ with discount factor $\rho$, $0 < \rho < 1$. Player 1 wishes to maximize the expected sum of his discounted rewards as player 2 wishes to minimize the same. In zero-sum games, an equilibrium point is called a saddle point; If it exists, a saddle point in stationary strategies $[\delta^*, \gamma^*]$ is a strategy pair such that, for any strategy pair $[\delta, \gamma]$ and for all $s \in S$,

$$v_{[\delta,\gamma^*]}(s) \leq v_{[\delta^*,\gamma^*]}(s) = v^*(s) \leq v_{[\delta^*,\gamma]}(s) \tag{1}$$

and the function $v^*$ is called the value of the game. Sufficient conditions for the existence of saddle points in continuous zero-sum games are given in [10].

Let $V$ represent the Banach space of all Borel-measurable bounded functions $v : S \rightarrow \mathbb{R}$, endowed with the supremum norm. In order to use a dynamic programming operators formalism, we define the local return function $h$ by

$$h(s,a,b) = r(s,a,b) + \rho \int_S v(s')q(ds'|s,a,b) \tag{2}$$

for $v \in V, s \in S, a \in A(s)$ and $b \in B(s)$. It represents the expected return to player 1 for a fictive auxiliary game starting in state $s$, if the players use the action pair $(a,b)$ and if the expected returns to player 1 from the next stage on are described by the function $v$. For every policy pair $[\delta, \gamma]$, the associated return operator $H_{[\delta,\gamma]} : V \rightarrow V$ is defined by:

$$H_{[\delta,\gamma]}(v)(s) = h(s, \delta(s), \gamma(s), v). \tag{3}$$

Finally, we define the operator $F : V \to V$ by

$$F(v)(s) = \sup_{a \in A(s)} ( \inf_{b \in B(s)} h(s, a, b, v)). \tag{4}$$

$H_{[\delta,\gamma]}$ and $F$ are both monotone contracting operators on $V$ with modulus $\rho$.

# 3 Value Iteration and Policy Iteration

Value iteration and policy iteration are two general methods for solving dynamic programs. They operate as follows.

**Value iteration.**
Select initial $v_0$ in $V$;
For $n := 1$ to $\bar{n}$ do

$$v_n := F(v_{n-1}); \tag{5}$$

Retain $[\delta^*, \gamma^*]$ such that $H_{[\delta^*,\gamma^*]}(v_{\bar{n}}) = F(v_{\bar{n}})$;
End.

**Policy iteration.**
Select initial policy pair $[\delta_0, \gamma_0]$;
For $n := 1$ to $\bar{n}$ do
Policy evaluation: find $v_n$ such that

$$H_{[\delta_{n-1},\gamma_{n-1}]}(v_n) = v_n; \tag{6}$$

Policy update: find $[\delta_n, \gamma_n]$ such that

$$H_{[\delta_n,\gamma_n]}(v_n) = F(v_n); \tag{7}$$

Retain $[\delta_{\bar{n}}, \gamma_{\bar{n}}]$;
End.

In both cases, the value of $\bar{n}$ may be chosen in advance or depend on some stopping criterion.

The algorithm proposed by Shapley [13] corresponds to value iteration. Each step requires the solution of $|S|$ matrix games in equation (5). It converges to $v^*$ from any starting $v_0$.

The algorithm proposed by Pollatschek and Avi-Ithzak [11] corresponds to policy iteration. Each step requires the solution of the system of $|S|$ linear equations (6) and $|S|$ matrix games (7). This algorithm does not converge in general for stochastic games.

It is well known that value iteration converges linearly (sometimes very slowly) while policy iteration (when it works) is equivalent to applying Newton's method to the equation $F(v) - v = 0$ (see [11]). When $v$ is not too far from $v^*$, it typically has quadratic convergence. Empirical evidence presented in [2] and [1] suggests that policy iteration is the fastest method for solving stochastic games in cases when it converges. Motivated by this fact, Filar and Tolwinsky [6] have recently proposed a modified Newton's method (MNM) which is guaranteed to converge and has the same rate of convergence as policy iteration when the latter converges.

In the context of MDPs having large state spaces, Puterman and Shin [12] proposed an adaptation of policy iteration, the so-called "modified policy iteration" method, where at each iteration, (6) is solved approximately by applying only a few iterations of the value iteration method with a fixed policy $[\delta_{n-1}, \gamma_{n-1}]$, starting from the previous $v$. Tolwinsky [14] proposed a modified iteration algorithm combining the MNM scheme with the ideas of [12]. In numerical experimentation, the modified policy iteration seemed to perform better than the MNM in cases where the number of states was large relative to the number of actions.

**Modified policy iteration.**
Select initial $v_0$ in $V$;
For $n := 1$ to $\bar{n}$ do
Policy update: find $[\delta_n, \gamma_n]$ such that

$$H_{[\delta_n,\gamma_n]}(v_n) = F(v_n); \tag{8}$$

Set $d := v_n$;
Search direction: select $k$ and repeat $k$ times

$$d := H_{[\delta_n,\gamma_n]}(d); \tag{9}$$

Choose a step size $\alpha$ ensuring descent and set $v_{n+1} := v_n + \alpha(d - v_n)$;
Retain $[\delta_{\bar{n}}, \gamma_{\bar{n}}]$;
End.

For a descent criterion in the modified Newton method, in order to choose the step size $\alpha$ according to Armijo's rule, see [6].

Obviously, for continuous (or very large) state spaces, these algorithms cannot be applied exactly in general. Some form of approximation must be used. From an approximate solution to the functional equation (5), one can obtain bounds on $v^*$ and on the suboptimality of a given policy pair (see [4]).

# 4 A finite Element Approach

We now introduce an approximate policy iteration algorithm, with finite element approximation of the value function. For more details on the finite element method, see e.g. [8]. Generally speaking, we assume that an expected "value-to-go" function $v$ associated with a fixed policy can be approximated reasonably well by a linear combination of a small set of (simple) base functions $w_1, \ldots, w_J$:

$$v(s) = \sum_{j=1}^{J} d_j w_j(s). \tag{10}$$

One particular finite element approach [9] is to select a finite number of points $\sigma_1, \ldots, \sigma_J$ in $S$ and to express directly $v(s)$ as a convex combination of the values of $v$ at the $J$ evaluation points:

$$v(s) = \sum_{j=1}^{J} v(\sigma_j) w_j(s) \tag{11}$$

where, for all $s \in S$,

$$0 \leq w_j(s) \leq 1, \tag{12}$$

$$w_j(\sigma_i) = \delta_{ij} \tag{13}$$

(the Kronecker's delta), and

$$\sum_j w_j(s) = 1. \tag{14}$$

The $\sigma_j$'s are in fact the *nodes* of the finite elements. The interesting point in this particular scheme is that it permits the evaluation of $v$ easily at any point in $S$, and thus on any set of nodes. In this setting, an analog to (9) is to apply pre-Jacobi iterations to the linear system

$$d = c + Md, \tag{15}$$

where $d$ and $c$ are the column vectors $(d_1, \ldots, d_J)'$ and $(c_1, \ldots, c_J)'$ respectively and $M$ is the $J \times J$ matrix $(m_{ij})$, with, for a given policy pair $[\delta, \gamma]$,

$$c_j = r(\sigma_j, \delta(\sigma_j), \gamma(\sigma_j)), \tag{16}$$

and

$$m_{ij} = \rho \int_S w_j(s') q(ds' | \sigma_i, \delta(\sigma_i), \gamma(\sigma_i)). \tag{17}$$

In general, policies must also be approximated: it is usually not possible to find $[\delta, \gamma]$ such that (5) is satisfied exactly when the action space is very large or continuous. As we did for the state space, we can define a finite dimensional subspace of the action spaces

$A$ and $B$ and consider only the actions that belong to that subspace. Since the detailed way to do that is rather problem-dependent, we will content ourselves with the following description. For any $v$ in $V$, $\epsilon \geq 0$ and $\sigma \in S^J$, define

$$\phi_\epsilon(v) = \left\{[\delta, \gamma] \in \Delta \times \Gamma : |H_{[\delta,\gamma]}(v) - F(v)| \leq \epsilon\right\} \tag{18}$$

and

$$\phi_\epsilon(v, \sigma) = \left\{[\delta, \gamma] \in \Delta \times \Gamma : |H_{[\delta,\gamma]}(v)(\sigma_i) - F(v)(\sigma_i)| \leq \epsilon\right\}. \tag{19}$$

At every "policy update" step of the algorithm (equation (8)), we will in fact seek a new policy in $\phi_\epsilon(v)$ for some given value of $\epsilon$. Often, in practice, we will first find a policy pair $[\delta, \gamma]$ and then estimate the smallest $\epsilon$ for which $[\delta, \gamma] \in \phi_\epsilon(v)$.

Under this setting, the approximation algorithm is given by:

**Approximate policy iteration**
Select $\epsilon > 0$, initial $v_0$ in $V$ and initial policy pair $[\delta_0, \gamma_0]$;
If average cost, select $\tilde{s} \in S$;
Outer loop: For $n := 1$ to $\bar{n}$ do
Select $J_n, \sigma = (\sigma_1, \ldots, \sigma_J)' \in S^J$ and $\{w_1, \ldots, w_J\} \subset V$
such that (12–14) are satisfied;
For the policy pair $[\delta_{n-1}, \gamma_{n-1}]$, compute $c$ and $M$ by (16–17)
and set $d := (v(\sigma_1), \ldots, v(\sigma_J))'$;
Inner loop (search direction): select $k$ and repeat $k$ times: $d := Md + c$;
Choose a step size $\alpha$ ensuring descent (see [6]) and set
$v_{n+1}(\sigma_i) := v_n(\sigma_i) + \alpha(d_i - v_n(\sigma_i))$;
Define $v$ by (11);
Select $\epsilon_1$ and find a new policy pair $[\delta_n, \gamma_n]$ in $\phi_{\epsilon_1}(v, \sigma)$;
If desired, perform a stopping test:
compute or estimate a bound $\bar{\epsilon}$ on $\|v^*, v_{[\delta_n,\gamma_n]}\|$.
If $\bar{\epsilon} \leq \epsilon$, or other stopping criteria satisfied, stop;
Endloop
End.

Obviously, as it stands, the approximation algorithm is not completely defined. For instance, the stopping criteria, the way of choosing $\epsilon, \bar{\epsilon}, \alpha, J$ and the base functions $w_j$, the method used to update the policy pair and to compute or estimate $\bar{\epsilon}$ are all left open. These are usually problem dependent. In practice, they may vary from iteration to iteration.

The stopping test can be costly and should not be performed at each iteration. The bound $\bar{\epsilon}$ may have to be estimated heuristically, for instance as in [7]: Recompute $F(v)$ and $H_{[\delta,\gamma]}(v)$ at a large nunber of new points, compute the approximation error at there points and take the largest and smallest to estimate the bounds.

Notice that for $k = 1$, the modified policy iteration method becomes the value iteration algorithm, as for $k = \infty$, it becomes policy iteration. A good choice for $k$ is probably problem dependent. It could be chosen adaptively, based on the previous iterations; intuitively, the more costly it is to compute $c$ and $M$, the larger the value of $k$ should be. But the inner loop should also stop when progress gets too slow, i.e. when $d$ is not changing significantly enough anymore of if $d$ does not appear to converge geometrically.

The choice of $\sigma$ determines a grid over the state space $S$. A coarser grid should be chosen at the early stages of the algorithm and the grid should be refined only when progress is stalling. Multigrid techniques [5] or various other techniques for the iterative solution of linear systems can also be used.

We have described a finite element approach to solve stochastic game models with continuous or very large state spaces. It can deal with most reasonably smooth value functions, provided that the state space is bounded and has few (continuous) dimensions. Numerical experiments with this approach are presently in progress.

# References


[1] Breton, M., Filar, J. A., Haurie, A. and Schultz, T. "On the Computation of Equilibria in Discounted Stochastic Games", in *Dynamic Games and Applications in Economics*, T. Başar Ed., Springer-Verlag, Berlin, (1986), 64–87.

[2] Breton, M. "Équilibres pour des jeux séquentiels", Ph. D. Thesis, Université de Montréal (1987).

[3] Breton, M. and L'Écuyer, P. "Noncooperative Stochastic Games Under a N-Stage Local Contraction Assumption", *Stochastics*, **26**, (1989), 227–245.

[4] Breton, M. "Algorithms for Stochastic Games", in *Stochastic Games and Related Topics - Shapley Honor Volume*, T. E. S. Raghavan, T. S. Ferguson, T. Parthasarathy and O. J. Vrieze Eds., Kluer, The Netherlands (to appear).

[5] Briggs, W. L. *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[6] Filar, J. A. and Tolwinsky, B. "On the Algorithm of Pollatschek and Avi-Itzhak", in *Stochastic Games and Related Topics - Shapley Honor Volume*, T. E. S. Raghavan, T. S. Ferguson, T. Parthasarathy and O. J. Vrieze Eds., Kluer, The Netherlands (to appear).

[7] Haurie, A. and L'Écuyer, P. "Approximation and Bounds in Discrete Event Dynamic Programming", *IEEE Transactions on Automatic Control*, **AC-31**, 3 (1986), 227–235.

[8] Hugues, T. J. R. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood, New Jersey, 1987.

[9] L'Écuyer, P. "Computing Approximate Solutions to Markov Renewal Programs with Continuous State Spaces", Technical Report DIUL-RR-8912, Université Laval, Québec, 1989.

[10] Nowak, A. S."On Zero-Sum Stochastic Games with General State Space I", *Probability and Mathematical Statistics*, **4**, (1984), 13–32.

[11] Pollatschek, M. and B. Avi-Itzhak "Algorithms for Stochastic Games with Geometrical Interpretation", *Management Science*, **15**, (1969), 399–415.

[12] Puterman, M. L. and Shin, M. C. "Modified Policy Iteration Algorithms for Discounted Markov Decision Problems", *Management Science*, **24**, 11 (1978), 1127–1137.

[13] Shapley, L. S. "Stochastic Games", *Proceedings of the National Academy of Sciences of USA*, **39**, (1953), 1095–1100.

[14] Tolwinsky, B. "Newton-Type Methods for Stochastic Games", in *Differential Games and Applications*, T. S. Başar and P. Bernhard Eds., Springer-Verlag, Berlin (1989), 128–143.

[15] Whitt, W. "Representation and Approximation of Noncooperative Sequential Games", *SIAM Journal on Control*, **18**, 1 (1980), 33–48.