Learning and Exploitation do not Conflict under Minimax Optimality*

Csaba Szepesvári

Research Group on Artificial Intelligence, "József Attila" University, Szeged, Aradi vrt. tere 1, Hungary, H-6720

Abstract. We show that adaptive real time dynamic programming extended with the action selection strategy which chooses the best action according to the latest estimate of the cost function yields asymptotically optimal policies within finite time under the minimax optimality criterion. From this it follows that learning and exploitation do not conflict under this special optimality criterion. We relate this result to learning optimal strategies in repeated two-player zero-sum deterministic games.

Keywords. *reinforcement learning, self-optimizing systems, dynamic games*

1 Introduction

Reinforcement learning (RL) concerns practical problems related to learning of optimal behaviour in sequential decision tasks. The most popular theoretical framework adopted by RL researchers is that of Markovian Decision Problems (MDPs). One of the main questions in RL is what extent of exploration is needed for a learner so that the price of exploration does not become too demanding. Usually some exploration (e.g. the execution of explorative actions that seem sub-optimal for the learner) is needed otherwise the learner may not gather some relevant information and this may eventually prevent convergence to optimality, i.e., the need for exploration and exploitation conflict.

In this paper we show that it is possible to obtain asymptotically optimal behaviour for MDPs *under the minimax discounted total cost criterion* with an algorithm that uses an *asynchronous, on-line* dynamic programming (DP)[1] iteration while executing *actions that seem optimal according to the actual state of the learner's knowledge*. Minimax sequential decision problems arise for example in repeated (or dynamic) games which are of great popularity in the machine learning community. Asynchronous on-line DP is a variant of value iteration which is advantageous when the state and action sets are so large that it is impractical to run a DP algorithm off-line, or when one is satisfied with obtaining an asymptotically optimal behaviour on some "relevant" part of the state space.

* This research was supported by OTKA Grant No. F20132. I would like to thank András Krámli for numerous helpful discussions which helped me to clarify the presentation of this paper substantially.

The paper is organized as follows: In Section 2 we provide the necessary background. In Section 3 we present important preliminary results concerning self-optimizing policies, the convergence of real-time DP and the main theorem. In Section 4 we discuss the results, and relate them to the context of learning to play repeated, two-player, zero-sum deterministic games. Due to space limits we can not present the proofs. The complete proofs and also the proofs of the unattributed statements can be found in [11, 10].

2 Background

We use the terminology of optimal sequential decisions. A 5-tuple $P = (X, A, \mathcal{A}, p, c)$ is called a finite Markovian Decision Process (homogeneous controlled Markov chain in the control terminology) if

1. both X and A are finite sets called the *state* space and the *action* space of the model, respectively
2. $\mathcal{A}$ maps the states into the power-set of A ($\mathcal{A} : X \to 2^A$). Elements of the set $\mathcal{A}(x) \subseteq A$ are called the *addmissible actions* for state x – for convenience we introduce the set $U = \{ (x, a) \in X \times A \mid a \in \mathcal{A}(x) \}$
3. $p : U \times X \to [0, 1]$ is the transition probability function which satisfies $\sum_{y \in X} p(x, a, y) = 1$ for all $(x, a) \in U$[2] and
4. $c : U \times X \to \mathbb{R}$ is the immediate cost function

At every stage of the decision process the decision maker observes one of the states of X (called the state of the decision maker) and is allowed to choose an action from $\mathcal{A}(x)$, which is then applied as an input to the system. As a result, if the state of the decision maker was $x \in X$ and action a was chosen then the next state of the decision maker becomes $y \in X$ with probability $p(x, a, y)$. Further, the application of action a in state x incurs an immediate cost $c(x, a, y)$.

The behaviour of the decision maker should be non-anticipating, i.e., it is assumed that it can be described by a (deterministic) policy $\pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$, where $\pi_t : (X \times A)^t \times X \to A$ determine the action to be taken at the tth time as a function of the history of the process $((x_0, a_0), (x_1, a_1), \ldots, (x_{t-1}, a_{t-1}))$ and the current state x_t. A policy is called Markovian if the actions taken by the policy depend only on the current state. A policy is stationary if it is Markovian and $\pi_t = \pi_0$ for all t.

Due to the stochastic nature of the decision problem P a policy π together with a starting state x_0 results in a stochastic process (Markovian, if the policy is Markovian) of form $\xi_t^\pi(x_0) = (x_t, a_t)$. The measure generated by $\xi_t^\pi(x_0)$ over the space of possible trajectories $U^{\mathbb{N}}$ is denoted by $\mu_\pi(x_0)$ [3].

For any trajectory of the decision process $\tau = (\tau_1, \tau_2, \ldots, \tau_t, \ldots)$, $\tau_t \in U$, let $X_{\text{i.o.}}(\tau)$ denote the states that are visited infinitely many times by the trajectory:

$$X_{\text{i.o.}}(\tau) = \{ x \in X \mid \tau_t = (x, a_t) \text{ infinitely often } \}.$$

[2] We indentify the set $U \times X$ with the set $\{ (x, a, y) \in X \times A \times X \mid (x, a) \in U, y \in X \}$.

Similarly, for all $x \in X_{\text{i.o.}}(\tau)$ let us define $A_{\text{i.o.}}{}^{\tau}(x) \subseteq \mathcal{A}(x)$ as the set of actions that are chosen infinitely many times during the trajectory τ:

$$A_{\text{i.o.}}{}^{\tau}(x) = \{\, a \in A \mid \tau_t = (x_t, a) \text{ infinitely often } \}.$$

The total cost incurred by the decision maker can be defined as follows: For any trajectory $\tau = ((x_0, a_0), (x_1, a_1), \ldots, (x_t, a_t), \ldots)$ of the process the immediate costs incurred during the process define the stochastic variable $C_\pi^\tau(x_0) = \sum_{t=0}^{\infty} \gamma^t c(x_t, a_t, x_{t+1})$, where $0 < \gamma < 1$ is a fixed discount factor. $C_\pi^\tau(x_0)$ is called the total discounted cost along trajectory τ. Since the variable $C_\pi^\tau(x_0)$ is random it does not determine an ordering of the policies – "decision functions", which map random variables to values must be used if one would like to arrive at an ordering [4]. One possibility is to use the ess sup operator when the cost of a policy π for a given starting state x_0 is determined as the worst probable value of $C_\pi^\tau(x_0)$:

$$v_\pi(x_0) = \sup_{\tau \in \text{support}(\mu_\pi(x_0))} C_\pi^\tau(x_0).$$

This gives rise to the minimax criterion, since the optimal cost funcion defined by $v_P^*(x) = \inf_\pi v_\pi(x)$ can now be written as

$$v_P^*(x) = \inf_\pi \sup_{\tau \in \text{support}(\mu_\pi(x_0))} C_\pi^\tau(x).$$

The optimal cost function can be shown to satisfy the Bellman Optimality Equation (BOE) [4, 11]:

$$v_P^*(x) = \min_{a \in \mathcal{A}(x)} \max_{(y \in X, p(x,a,y) > 0)} \left(c(x,a,y) + \gamma v_P^*(y) \right), \qquad x \in X. \tag{1}$$

Since this criterion does not depend on the exact numerical values of the transition probabilities we introduce the sets

$$T(x,a) = \{\, y \in X \mid p(x,a,y) > 0 \,\}, \quad (x,a) \in U$$

and for convenience we will call the 5-tuple $P = (X, A, \mathcal{A}, T, c)$ a Markovian Decision Process (MDP), too. *In the following a MDP will be understood with the minimax criterion.*

Once the optimal cost function is identified, the optimal policies of the decision maker are easy to determine: the possible optimal choices for state x are given as the elements of the set $A^P(x, v_P^*)$, where

$$A^P(x,v) = \operatorname*{Argmin}_{a \in \mathcal{A}(x)} \max_{y \in T(x,a)} \left(c(x,a,y) + \gamma v(y) \right), \qquad v \in B(X).$$

Elements of $A^P(x,v)$ are called greedy w.r.t. v in state x.

Let $B(X)$ denote the set of real-valued bounded functions over the set X and let the operator $G : B(X) \to B(X)$ be defined by

$$(Gf)(x) = \min_{a \in \mathcal{A}(x)} \max_{y \in T(x,a)} \left(c(x,a,y) + \gamma f(y) \right), \qquad x \in X.$$

Using G, Equation (1) takes the shorter form $Gv^*_P = v^*_P$, i.e., v^*_P is the fixed point of G. It is easy to prove that G is a contraction with index γ w.r.t. the supremum norm $\|\cdot\|_\infty$, and therefore it has only one fixed point and the sequence $v_{t+1} = Gv_t$, where $v_0 \in B(X)$ is arbitrary, converges to v^*_P. This iteration is called the (synchronous) dynamic programming (SDP) iteration. If it is important to indicate the problem underlying the greedy operator G (often also called the value operator) then this will be denoted by a subscript, such as G_P. If we do not use the subscript, this means that the greedy operator that corresponds to P (the true model of the system under control) is considered.

In this article we consider the adaptive algorithm given in Table 1. The algorithm consists of 3 main parts: a model building part (Step 2), a cost-function estimation part (Step 3 and 4), and the action selection procedure (Step 5). *The algorithm always chooses the greedy action w.r.t. the latest estimate of the cost function.* We will prove that despite this non-explorative choice of actions, the actions generated by this algorithm become optimal in finite time for almost all trajectories, i.e., this algorithm is *eventually optimal*.[3] It is an open question if the expected number of steps until optimality, $E[T]$, is bounded or not. The pointwise contraction property of the greedy operator may provide a starting point to prove the boundedness of $E[T]$.

3 Results

The adaptive algorithm is greatly simplified if one uses the initialization $T_0(x,a) = T(x,a)$ and $c_0(x,a) = c(x,a)$, $(x,a) \in U$ in Step 0 of Algorithm 1 (which means that the decision model is known to the learner). The corresponding algorithm is called the Myopic Real-Time Dynamic Programming (MRTDP) algorithm. Of course, there is no need for Step 2 in this case, since $T_t = T$ and $c_t = c$ for each t. The proof of the convergence of the adaptive version is reduced to the convergence of the non-adaptive version.

Theorem 1 MARTDP Convergence Theorem. *Let $P = (X, A, \mathcal{A}, T, c)$ be an arbitrary MDP where $c \geq 0$. Let $v_0 = 0$ and assume that $c_0 \leq c$ (e.g., $c_0 = 0$). Assume that $\{x_t\} \subseteq F_t$ for all $t = 1, 2, 3, \ldots$. Then for almost all trajectory τ with starting state x_0, v_t converges to v^*_P on the set of states visited infinitely often by τ:*

$$v^*_P|_{X_{\mathrm{i.o.}}(\tau)} = \lim_{t\to\infty} v_t|_{X_{\mathrm{i.o.}}(\tau)},$$

Further, for these trajectories the algorithm results in an optimal strategy in finite time, i.e., there exists a (random) time T such that if $t > T$ then $a_t \in A^P(x_t, v^*_P)$.

The proof is carried out in several steps. First one proves the theorem in the special case when the immediate costs are known in advance. The idea of this proof is that for any pair $(x,a) \in U$ the set $T_t(x,a)$ monotonically increases in time and since U is finite there exists a time T_0 such that if $t > T_0$ then

[3] Note that the time after which the actions generated by the algorithm are optimal depend on the actual trajectory, i.e., it is a stopping time.

0. Let $T_0(x,a) \subseteq T(x,a), c_0(x,a,y) \leq c(x,a,y)$ and $v_0 \leq v_P^*, v_0 \in B(X)$.
repeat forever{
1. Observe the current state x_t and the reinforcement signal c_t.
2. Modify the model $P_t = (X, A, \mathcal{A}, T_t, c_t)$:

$$T_{t+1}(x,a) = \begin{cases} T_t(x,a) \cup \{x_t\}, & \text{if } (x,a) = (x_{t-1}, a_{t-1}); \\ T_t(x,a), & \text{otherwise,} \end{cases}$$

$$c_{t+1}(x,a,y) = \begin{cases} \max(c_t, c_t(x,a,y)), & \text{if } (x,a,y) = (x_{t-1}, a_{t-1}, x_t); \\ c_t(x,a,y), & \text{otherwise.} \end{cases}$$

3. Select the set of states $F_t \subseteq X$ whose estimated cost is to be refreshed.
4. Refresh the cost function estimates:

$$v_{t+1}(x) = \begin{cases} v_t(x), & \text{if } x \notin F_t; \\ (G_{P_{t+1}} v_t)(x), & \text{if } x \in F_t; \end{cases}$$

where $G_{P_{t+1}}$ is the actual estimate of the greedy operator G based on P_{t+1}.
5. Action selection: $a_t \in A^{P_{t+1}}(x_t, v_{t+1})$.
6. Execute the action.
7. $t := t+1$.
}

Table 1. Myopic Adaptive Real-time Dynamic Programming (MARTDP).

In Step 4 the update of a state x, for which there is no non-empty $T_t(x,a)$, should be understood as the identity operation.

$T_t(x,a) = T_{t+1}(x,a) = T_\infty(x,a)$. It is also clear that $T_\infty(x,a) \subseteq T(x,a)$. If $t > T_0$ then Algorithm 1 thus acts like a myopic RTDP algorithm corresponding to the problem $P_\infty = (X, A, \mathcal{A}, T_\infty, c)$.

The convergence of MRTDP can be shown in the following way: For almost all trajectories of the process τ, v_t can be shown to converge to some function $v = v_\tau \in B(X)$ which satisfies the BOE of P on $X_{\text{i.o.}} = X_{\text{i.o.}}(\tau)$:

$$v(x) = \min_{a \in \mathcal{A}(x)} \max_{y \in T(x,a)} (c(x,a,y) + \gamma v(y)), \quad x \in X_{\text{i.o.}} \tag{2}$$

and if $x \in X_{\text{i.o.}}(\tau)$ then $A_{\text{i.o.}}{}^\tau(x) \subseteq A^P(x,v)$. From these it follows that $v|_{X_{\text{i.o.}}(\tau)} = v_P^*|_{X_{\text{i.o.}}(\tau)}$ and by the following Lemma the eventual optimality of the MRTDP algorithm.

Lemma 2. *Let $P = (X, A, \mathcal{A}, T, c)$ be an arbitrary MDP and π a policy. Let $x_0 \in X$ be fixed. The following statements are equivalent:*

(i) π is eventually optimal, i.e., for almost all trajectory τ generated by π and x_0 there exist a time T such that if $t > T$ then $a(\tau, t)$ is optimal.

(ii) For almost all trajectory τ the optimal cost function of P satisfies

$$v_P^*(x) = \min_{a \in A_{\text{i.o.}}{}^\tau(x)} \max_{y \in T(x,a)} (c(x,a,y) + \gamma v_P^*(y)), \quad x \in X_{\text{i.o.}}(\tau). \tag{3}$$

If the costs are not known in advance then we have a sequence of cost functions, c_t, which is monotonically increasing and (since they are bounded by c) converges to some function c_∞. The rest of the proof follows almost the same lines as the proof when the costs are known in advance. The complete proof can be found in [10].

Note that in order to prove the convergence of the adaptive version we had to prove the convergence of the non-adaptive version. We could not find any simple way to extend the convergence proof of the RTDP theorem by Barto et al. [1] or that of the LRTA* algorithm by Korf [5] to our case. The main differences are that Barto et al. assume that $F_t = \{x_t\}$ in each step which makes it possible to reduce the convergence of the RTDP algorithm to that of the asynchronous dynamic programming in a direct way and that Korf assumed deterministic transitions which enables a non-probabilistic approach simplifying the proof greatly.

4 Consequences for repeated two-player zero-sum deterministic games

We show that repeated two-player zero-sum deterministic games can be viewed as minimax control problems. In such a game one or the other player chooses an action in each state with one player striving to minimize the total cost and the other trying to maximize it. A great deal of RL research has been directed to solving games of this kind [13, 14, 7, 2, 12, 6]. It is possible to define a set of Bellman equations for the optimal minimax value of such a game,

$$V^*(x) = \begin{cases} \min_{a \in A} \left(C(x,a) + \gamma V^*(t(x,a)) \right), & \text{if minimizer moves in } x \\ \max_{a \in A} \left(C(x,a) + \gamma V^*(t(x,a)) \right), & \text{if maximizer moves in } x, \end{cases}$$

where $C(x,a)$ is the cost incurred by the minimizing player (and is the reward of the maximizing player since the game is zero sum in each step) and $t(x,a)$ is the next state when action a is executed from x [8]. Without loosing generality one may assume that the two players move one another in each step, i.e., the move of player I is always followed by the move of player II, and vice versa. Now, it is easy to see that the restriction of V^* to those states in which the minimizer moves satisfies the equality $V^*(x) = \min_{a \in A} \max_{y \in T(x,a)} (c(x,a,y) + \gamma^2 V^*(y))$, where $T(x,a) = \{\, t(t(x,a),b) \,|\, b \in A\,\}$ and $c(x,a,y) = C(x,a) + \gamma \max_{b: t(t(x,a),b)=y} C(t(x,a),b)$. This shows that from the point of view of the minimizer the game is of minimax nature.

If the rules of the game are known to the minimizer, i.e., she has access to c and T, then it follows from our results that her policy will become eventually optimal if she plays according to the MRTDP algorithm. It is interesting to note that if the minimizer plays with a maximizer who uses a fixed (although arbitrary) policy then she can do even better if he uses the adaptive version of the MRTDP algorithm. If the maximizer uses an eventually sub-optimal policy then it may happen that $T_\infty(x,a)$ as learned by the MARTDP algorithm will be a proper subset of $T(x,a)$. The optimal cost function associated with the MDP

$(X, A, \mathcal{A}, T_\infty, c)$ may be smaller then V^*, meaning that the adaptive strategy allows the minimizer to take advantage of the sub-optimality of the policy of the maximizer. However, if the maximizer eventually finds the optimal counter-policy then the "adaptive" minimizer may perform worse in terms of the total regret than the non-adaptive version. If both the minimizer and the maximizer use the MARTDP algorithm (the above proofs can be trivially extended to maximin models) then both player will eventually play the optimal policies. In other words, if an agent trained by MARTDP plays against herself then she is guaranteed to find the optimal policy eventually. We have compared the learning curves of ARTDP algorithms with different action selection strategies on the game 'Tic-Tac-Toe'. The results (to be presented elsewhere) show that the myopic strategies (i.e., the MARTDP algorithm) yields the fastest convergence.

It is important to note that our result about the MARTDP algorithm does *not* extend to the case of stochastic games with the *expected* total discounted cost criterion. The reason of this can be understood by considering the following simple case: assume that the state space has only one element and we have two actions, a and b. Let the cost of the actions be random with finite expected values $Q^*(b) < Q^*(a)$ (this problem can also be given in terms of deterministic costs and stochastic transitions over an augmented state space). The problem is the action with the minimal cost. The MARTDP procedure (with the greedy action selection strategy) will choose the action with the smallest estimated cost, i.e., it chooses a if $Q_t(a) < Q_t(b)$. If $Q_0(b) > Q_t(a)$ for all t then the algorithm will fail to find the optimal action, b. This situation may happen with *positive probability* if $Q_0(b) > Q^*(b)$. One may notice that the conditions of Theorem 1 outrule this possibility. Now, if $Q_0 \leq Q^*$ for both a and b, then due to the non-zero variance of the random cost it may still happen with positive probability that $Q_t(b) > Q^*(b) > Q_t(a)$ for some t, and thus we find again that the MARTDP algorithm fails with positive probability. Another problem arises when just part of the state space (in fact state-action space) is visited infinitely often. For expected value models improperly estimated values of finitely visited states may result in improper estimation of infinitly often visited states.

If the stochasticity in the effect of actions is caused by noise then the minimax optimality criterion will tend to find overly "pessimistic" policies which may be undesirable. One may weaken this effect by introducing a positive threshold θ and letting $T(x,a) = \{y \in X \mid P(x,a,y) > \theta\}$. In the adaptive case this calls for the identification of the transition probabilities. Because of the stochasticity of the identification process the inclusions $T_t(x,a) \subseteq T(x,a)$ need no longer holds. This may prevent convergence to the optimal policy. To overcome this problem we propose to introduce state, action and time dependent thresholds, $\theta_t(x,a)$ which should converges to θ from above slowly. We conjecture that by either using the law of iterated logarithm or a two-stage estimation procedure as in [9] asymptotic convergence to an optimal policy can be retained.

References

1. A.G. Barto, S.J. Bradtke, and S.P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:91–138, 1995. Technical Report 91-57, Computer Science Department, University of Massachusetts, Vol. 59., 1991.
2. Justin A. Boyan. *Modular Neural Networks for Learning Context-Dependent Game Strategies.* Master's thesis, Department of Engineering and Computer Laboratory, University of Cambridge, Cambridge, UK, August 1992.
3. E.B. Dynkin and A.A. Yushkevich. *Controlled Markov Processes.* Springer-Verlag, Berlin, 1979.
4. M. Heger. *Risk-sensitive decision making.* PhD thesis, Zentrum für Kognitionwissenschaften, Universität Bremen, FB3 Informatik, Postfach 330 440, 28334 Bremen, Germany, 1996.
5. R.E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42:189–211, 1990.
6. M.L. Littman and Cs. Szepesvári. A Generalized Reinforcement Learning Model: convergence and applications. In *Int. Conf. on Machine Learning*, 1996. http://iserv.iki.kfki.hu/asl-publs.html.
7. Nicol N. Schraudolph, Peter Dayan, and Terrence J. Sejnowski. Using the TD(λ) algorithm to learn an evaluation function for the game of Go. In *Advances in Neural Information Processing Systems 6*, Morgan Kaufmann, San Mateo, CA, 1994.
8. L.S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, 39:1095–1100, 1953.
9. C. Stein. A two-sample test for a linear hypothesis whose power is independent of variance. *Ann. Math. Statist.*, 16, 1945.
10. Cs. Szepesvári. *Certainty equivalence policies are self-optimizing under minimax optimality.* Technical Report 96-101, Research Group on Artificial Intelligence, JATE-MTA, Szeged 6720, Aradi vrt tere 1., HUNGARY, August 1996. URL: http://www.inf.u-szeged.hu/~rgai.
11. Cs. Szepesvári. *Some basic facts concerning minimax sequential decision problems.* Technical Report 96-100, Research Group on Artificial Intelligence, JATE-MTA, Szeged 6720, Aradi vrt tere 1., HUNGARY, August 1996. URL: http://www.inf.u-szeged.hu/~rgai.
12. Cs. Szepesvári and M. Littman. Generalized Markov Decision Processes: Dynamic programming and reinforcement learning algorithms. *Operations Research*, 1996. in preparation.
13. Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 58–67, March 1995.
14. Sebastian Thrun. Learning to play the game of chess. In *Neural Information Processing Systems 7*, 1995.

This article was processed using the LaTeX macro package with LLNCS style