A Q-LEARNING ALGORITHM WITH CONTINUOUS STATE SPACE

KENGY BARTY, PIERRE GIRARDEAU, JEAN-SÉBASTIEN ROY, AND CYRILLE STRUGAREK

Abstract. We study in this paper a Markov Decision Problem (MDP) with continuous state space and discrete decision variables. We propose an extension of the Q-learning algorithm introduced to solve this problem by Watkins in 1989 for completely discrete MDPs.

Our algorithm relies on stochastic approximation and functional estimation, and uses kernels to locally update the Q-functions. We give a convergence proof for this algorithm under usual assumptions. Finally, we illustrate our algorithm by solving the classical moutain car task with continuous state space.

1. Introduction

In a Markov Decision Problem (MDP), an agent wanders in a markovian environment and tries to maximize its expected long-term reward (or to minimize its long-term cost), by performing actions that have to depend only on the current state.

Simple examples of MDPs are concerned with leading an agent moving on a surface to a certain goal in shortest time, when its trajectory may be affected by some sort of deterministic or stochastic process (wind for example). More complicated tasks may be written using the mathematical model of MDPs, such as controlling an hydro-power plant that has to satisfy a demand over a certain period of time, while minimizing the cost of the thermal power production if the hydro-power plant cannot supply the demand completely.

Dynamic programming is a powerful methodology for dealing with sequential decision making problems under uncertainty like MDPs. In the case of a continuous state space, the usual approach is to discretize the state space and to apply recursively the Bellman operator. This discretization usually leads to very large state spaces. It is known as the curse of dimensionality. An additional complexity arises in the stochastic case, since the conditional expectation appearing in the Bellman equation must also be approximated through a discretization of the dynamics.

However, in the MDP setting, reinforcement learning combined with the theory of dynamic programming led to very efficient algorithms in the case of a discrete state, via the TD(λ) algorithm of Sutton [Sut88] and the Q-learning algorithm of Watkins [Wat89]. Moreover, it is proved that Q-learning [Wat89, WD92] and TD(λ) [Tsi93, JJS94] algorithms converge with probability one.

Unfortunately, in the case where the state space is continuous, discretizing can only lead to near-optimal solutions. Ormoneit et al. [OS99] [OG00] recently proposed to estimate the value functions using non-parametric regression methods, such as kernel-based methods. They showed that their algorithm could be applied even when classical algorithms based on discretization of the state space failed to converge. A major drawback is that the method is not recursive: it approximates the value function using estimation points, and when one wants to increase the number of estimation points, the previous estimate cannot be used to derive the new one.

We present an algorithm that extends Q-learning to the case of a continuous state space, by using local updates with kernels to estimate the value functions. Our method is recursive and non-parametric. It is based on stochastic approximation (see [RM51], or [Lai03] for an historical survey of these techniques). Since we avoid the space discretization, our method leads to the optimal solution of the original problem. Moreover, it is convenient from a practical point of view since it avoids discretizing the dynamics.

In section 2, we present the Q-learning formulation. Then we introduce the Robbins-Monro and the TD(0) algorithms, which are closely related to our proposed method. In the same section, we present our kernel method and provide a convergence proof for this algorithm under assumptions that are classical in

Date: September 22, 2006.

stochastic approximation. Then, in section 3, we solve the mountain car task with the newly presented algorithm.

2. Theoretical Framework

2.1. **The Q-learning formulation.** Let us consider an optimal control problem with a finite set of decision dates $\{0, 1, \dots, T\}$. The uncertainties W are supposed to be independent random variables $(W_j)_{j\in\{0,1,\dots,T\}}$ with values in $\mathbb{R}^m$. Let $(X_j)_{j\in\{0,1,\dots,T\}}$ be the controlled random process with values in $S \subseteq \mathbb{R}^d$ defined by:

$$X_{j+1} = f_j\left(X_j, \gamma_j\left(X_j\right), W_{j+1}\right), \forall j \in \{0, \dots, T-1\},$$

with: $\gamma_j : S \to \mathbb{U}_j^{ad}$ the decision rule.

We denote by $L_j\left(X_j, \gamma_j\left(X_j\right), W_{j+1}\right)$ the cost at state X_j, when action $\gamma_j\left(X_j\right)$ is taken, and uncertainty W_{j+1} occurs. We also consider a cost at final time denoted $G\left(X_T\right)$. Our aim is to minimize the expected global cost, starting from a state $x \in S$, i.e.:

$$V_0^*(x) = \min_{(\gamma_j)_{j=0,\dots,T}} \mathbb{E}\left[\sum_{j=0}^{T-1} L_j\left(X_j, \gamma_j\left(X_j\right), W_{j+1}\right) + G\left(X_T\right) \middle| X_0 = x\right], \quad \text{with } \gamma_j : S \to \mathbb{U}_j^{ad}, \quad \forall j = 0, \dots, T. \tag{1}$$

Now let us note $V_j^*(x)$ the expectation of the future cost under the optimal control policy, starting from state x at time j. Equation (1) can be rewritten as a dynamic programming equation:

$$\begin{cases} V_j^*(x) = \min_{u\in\mathbb{U}_j^{ad}} \mathbb{E}\left[L_j\left(x, u, W_{j+1}\right) + V_{j+1}^*\left(f_j\left(x, u, W_{j+1}\right)\right)\right], \forall j \in \{0, \dots, T-1\}, \forall x \in S \\ V_N^*(x) = G(x), \forall x \in S. \end{cases} \tag{2}$$

We now present the Q-learning counterpart of equation (2). Let us denote by $Q_j^*(x, u)$ the expectation of the future cost starting from state x and taking action u at time j. We have the following relation between Q_j^* and V_{j+1}^*:

$$Q_j^*(x, u) = \mathbb{E}\left[L_j\left(x, u, W_{j+1}\right) + V_{j+1}^*(x)\right], \forall j \in \{0, \dots, T-1\}.$$

Moreover, Q_j^* can be derived from Q_{j+1}^* as follows:

$$\begin{cases} Q_j^*(x, u) = \mathbb{E}\left[L_j\left(x, u, W_{j+1}\right) + \min_{v\in\mathbb{U}_{j+1}^{ad}} Q_{j+1}^*\left(f_j\left(x, u, W_{j+1}\right), v\right)\right], \forall j \in \{0, \dots, T-1\}, \\ \qquad\qquad \forall x \in S, \forall u \in \mathbb{U}_j^{ad} \\ Q_N^*(x, u) = G(x), \forall x \in S. \end{cases} \tag{3}$$

Equation (3) can be seen as a fixed point equation on $Q^* = (Q_t^*)_{0\le t\le T}$.

2.2. **The Robbins-Monro and TD(0) algorithms.** We now consider the feasible decision sets $\left(\mathbb{U}_j^{ad}\right)_{j=0,\dots,T}$ to be finite. At this step, the control policy γ_j at each time j is still a mapping from the state space S into the feasible control set $\mathbb{U}_j^{ad}$. Two classical approaches apply in the case where the state space S is discrete and not too large: The Robbins-Monro and the TD(0) algorithms.

In order to estimate $Q_j^*\left(x, u\right)$, Robbins and Monro [RM51] introduced the following algorithm:

$$Q_j^{k+1}\left(x, u\right) = Q_j^k\left(x, u\right) + \rho^{k+1}\, \Delta_j^{k+1}\left(x, u\right), \quad \forall\left(x, u\right) \in S \times \mathbb{U}_j^{ad},$$
$$\text{where } \Delta_j^{k+1}\left(x, u\right) = \left(L_j\left(x, u, W_{j+1}^{k+1}\right) + \min_{v\in\mathbb{U}_{j+1}^{ad}} Q_{j+1}^k\left(f_j\left(x, u, W_{j+1}^{k+1}\right), v\right)\right) - Q_j^k\left(x, u\right),$$

and W_{j+1}^{k+1} is a realization of the process W_{j+1}. Note that at each iteration the update is performed for every state and action (x,u), and each time step.

Instead of updating Q_j for every state and action (x,u), Sutton [Sut88] proposed to randomize this operation by drawing realizations X_j^k of the random variable X_j. The finite control space $\mathbb{U}^{ad}$ is randomly explored along the iterations as well by drawing possible decisions u_i according to probabilities π_{u_i} so that $\mathbb{P}(u=u_i)=\pi_{u_i}, \forall u_i \in \mathbb{U}^{ad}$. We enforce $\pi_{u_i}>0, \forall u_i \in \mathbb{U}^{ad}$, so that, with an infinite number of iterations, every possible strategy will be tested with probability 1. We hence obtain the TD(0) algorithm:

$$Q_j^{k+1}(x,u)=\begin{cases} Q_j^k\left(X_j^k,U_j^k\right)+\rho_j^{k+1}\,\Delta_j^{k+1}\left(X_j^k,U_j^k\right) & \text{if } \left(X_j^k,U_j^k\right)=(x,u),\\ Q_j^k(x,u) & \text{else.}\end{cases}$$

Unfortunately, this algorithm cannot be implemented if the state space is continuous and is untractable if the state space is discrete but too large: the computational burden would be too important.

2.3. **Q-learning with kernels.** We propose an alternative approach that is non-parametric and avoids any a priori discretization of the state space. However, control spaces $\left(\mathbb{U}_j^{ad}\right)_{j=0,\dots,T}$ are still assumed discrete. As in the TD(0) algorithm, we draw at each step of the algorithm realizations of the state x and the control u so that $\mathbb{P}(u=u_i)=\pi_{u_i}, \forall u_i \in \mathbb{U}^{ad}$.

Our algorithm consists in replacing the pointwise updates in the TD(0) algorithm by local updates with kernels K^k, whose bandwidths ε^k decrease along the iterations, using a well-known analysis result, for kernels having certain properties, and for any function f regular enough (see e.g. [Boc55], Theorem 1.3.2):

$$f(\cdot)=\lim_{k\to+\infty}\mathbb{E}\left(f(X)\,\frac{1}{\varepsilon^k}K^k(X,\cdot)\right).$$

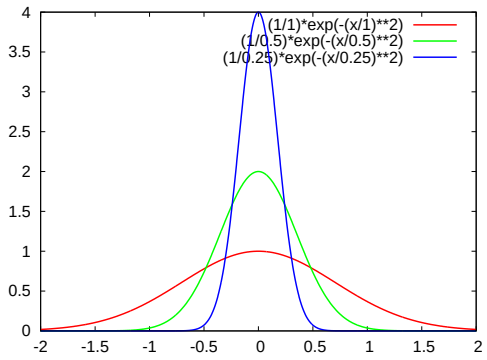

FIGURE 1. Gaussian kernels with several bandwidths.

This idea could a priori be applied to both the state x and the control u. However, the minimization operation at each iteration $\min_{v\in\mathbb{U}_{j+1}^{ad}} Q_{j+1}^k\left(f_j\left(x,u,W_{j+1}^{k+1}\right),v\right)$ would become hard to perform, since function Q_{j+1}^k would not be convex in the control u. Hence it appears natural to consider a discrete control space, in which case the minimization above can be performed more easily.

Our algorithm reads as follows:

Algorithm 2.1. *Initialize $Q_{j,u}^0(\cdot,\cdot)$ to 0 for all $j\in\{0,\dots,T-1\}$,*
Step $k+1, k\geq 0$:

- *Draw $\left(W_j^{k+1}\right)_{1\leq j\leq T}$ independently from the past drawings, then draw $U^{k+1}=\left(U_j^{k+1}\right)_{0\leq j\leq T-1}$ as described above and finally compute $X^{k+1}=\left(X_j^{k+1}\right)_{0\leq j\leq T}$ according to:*

$$X_{j+1}^{k+1}=f_j\left(X_j^{k+1},U_j^{k+1},W_{j+1}^{k+1}\right).$$

- *Update the value functions* $Q_j^{k+1}(\cdot,\cdot)$ *in a neighbourhood of the drawings* $\left(X_j^{k+1}\right)_{0\leq j\leq T}$ *:*

$$\left\{\begin{array}{rcl} Q_T^k(x,u) & = & G(x), \quad \forall x,u, \\ Q_{T-1}^{k+1}(\cdot,\cdot) & = & Q_{T-1}^k(\cdot,\cdot)+\rho^{k+1}\,\Delta_{T-1}^{k+1}\,K_{T-1}^{k+1}(X_{T-1}^{k+1},U_{T-1}^{k+1},\cdot,\cdot), \\ & \vdots & \\ Q_j^{k+1}(\cdot,\cdot) & = & Q_j^k(\cdot,\cdot)+\rho^{k+1}\,\Delta_j^{k+1}\,K_j^{k+1}(X_j^{k+1},U_j^{k+1},\cdot,\cdot), \\ & \vdots & \\ Q_0^{k+1}(\cdot,\cdot) & = & Q_0^k(\cdot,\cdot)+\rho^{k+1}\,\Delta_0^{k+1}\,K_0^{k+1}(X_0^{k+1},U_0^{k+1},\cdot,\cdot). \end{array}\right.$$

where, for all $j\leq T-1$*:*

$$\Delta_j^{k+1}=\left(L_j\left(X_j^{k+1},U_j^{k+1},W_{j+1}^{k+1}\right)+\min_{v\in\mathbb{U}_{j+1}^{ad}}Q_{j+1}^k\left(\underbrace{f_j\left(X_j^{k+1},U_j^{k+1},W_{j+1}^{k+1}\right)}_{X_{j+1}^{k+1}},v\right)\right)-Q_j^k\left(X_j^{k+1},U_j^{k+1}\right).$$

Functions K_j^k are kernels, i.e. bounded mappings. A typical choice for these mappings is Gaussian function (see figure 1):

$$K_j^{k+1}(X_j^{k+1},U_j^{k+1},x,u)=\delta_{\{U_j^{k+1}=u\}}e^{-\left\|\frac{x-X_j^{k+1}}{\eta^k}\right\|^2},$$

where $\eta^k\to 0$ when $k\to+\infty$.

2.4. **Convergence Proof.** Since we draw the control u independently from the number of iterations k, X_j follows a law of probability that is independant of k. Thus, we can define the following inner products and norms:

$$\forall j=0,\ldots,T-1,\langle g,h\rangle_{\mu_j}:=\mathbb{E}\left[\sum_{i=1}^{card(\mathbb{U}^{ad})}\pi_{u_i}g\left(X_j,u_i\right)h\left(X_j,u_i\right)\right],$$

$$\|e\|_{\nu_j}:=\mathbb{E}\left[\left\langle e\left(f_{j-1}\left(\cdot,\cdot,W_j\right)\right),e\left(f_{j-1}\left(\cdot,\cdot,W_j\right)\right)\right\rangle_{\mu_j}\right],$$

Moreover, we introduce:

$$v_{j+1}^k\left(f_j\left(x,u,W_{j+1}^{k+1}\right)\right)=\arg\min_{v\in\mathbb{U}_{j+1}^{ad}}Q_{j+1}^k\left(f_j\left(X_j^{k+1},U_j^{k+1},W_{j+1}^{k+1}\right),v\right),$$

and:

$$r_j^k(x,u)=\mathbb{E}^k\left[L_j\left(x,u,W_{j+1}^{k+1}\right)+Q_{j+1}^k\left(f_j\left(x,u,W_{j+1}^{k+1}\right),v_{j+1}^k\left(f_j\left(x,u,W_{j+1}^{k+1}\right)\right)\right)\right]-Q_j^k(x,u).$$

Finally, we denote by $V_j^k(x):=\min_{u\in U^{ad}}Q_j^k(x,u)$ the k-th approximation of the Bellman function on x.

Theorem 2.2. *If, for all* $j\in\{0,\ldots,T\}$*, there exists* $b_j\in\mathbb{R}$ *such that :*

(4a)

$$\left\|r_j^k(\cdot,\cdot)-\mathbb{E}^k\left[\Delta_j^{k+1}\frac{1}{\varepsilon_k}K^k\left(X_j^{k+1},U_j^{k+1},\cdot,\cdot\right)\right]\right\|_{\mu_j}\leq b_j\varepsilon_k\left(1+\left\|V_{j+1}^k-V_{j+1}^*\right\|_{\nu_{j+1}}+\left\|Q_j^k-Q_j^*\right\|_{\mu_j}\right),$$

(4b)
$$\sum_{k=1}^{+\infty}\rho_k^2\mathbb{E}^k\left[\left\|\Delta_j^{k+1}K^k\left(X_j^{k+1},U_j^{k+1},\cdot,\cdot\right)\right\|_{\mu_j}^2\right]<+\infty,$$

(4c)
$$\varepsilon_k\xrightarrow[k\to+\infty]{}0,\quad\sum_{k\in\mathbb{N}}\rho_k\varepsilon_k^2<+\infty,\quad\sum_{k\in\mathbb{N}}\rho_k\varepsilon_k=+\infty,$$

then Q-functions $Q_j^k(\cdot,\cdot)$ *defined by Algorithm 2.1 converge a.s., when* $k\to+\infty$*, toward the solution* $\left(Q_j^*(\cdot,\cdot)\right)_{j=0,\ldots,T}$ *of equation (3).*

Proof : For $j\in\{0,\ldots,T-1\}$, we introduce $a_j^k=\left\|Q_j^k-Q_j^*\right\|_{\mu_j}^2$. We are going to prove the following property for all $j\in\{0,\ldots,T\}$:

(P_j)
$$\sum_{k=1}^{+\infty}\rho_k\varepsilon_k\left\|V_j^k-V_j^*\right\|_{\nu_t}^2<+\infty.$$

Simultaneously, we are going to prove that for all $j \in \{0, \dots, T-1\}$ one has:

$$\lim_{k \to +\infty} a_j^k = 0$$

1. Since the Bellman function V_N^* is perfectly determined via the final condition of the dynamic programming equation (2), one has that property (P_N) is verified.

2. Now suppose (P_{j+1}) is true. Then, by Pythagore's theorem:

$$a_j^{k+1} = a_j^k + 2\rho_k \varepsilon_k \underbrace{\left\langle \frac{\Delta_j^{k+1}}{\varepsilon_k} K^k \left(X_j^{k+1}, U_j^{k+1}, \cdot, \cdot \right), Q_j^k - Q_j^* \right\rangle_{\mu_j}}_{A} + \rho_k^2 \left\| \Delta_j^{k+1} K^k \left(X_j^{k+1}, U_j^{k+1}, \cdot, \cdot \right) \right\|_{\mu_j}^2,$$

Let us take the conditional expectation of A:

$$\begin{aligned}
\mathbb{E}^k [A] &= \left\langle \mathbb{E}^k \left[\frac{\Delta_j^{k+1}}{\varepsilon_k} K^k \left(X_j^{k+1}, U_j^{k+1}, \cdot, \cdot \right) \right] - r_j^k (\cdot, \cdot), Q_j^k - Q_j^* \right\rangle_{\mu_j} \\
&\quad + \left\langle r_j^k (\cdot, \cdot), Q_j^k - Q_j^* \right\rangle_{\mu_j}, \\
&\leq b\varepsilon_k \left(1 + \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_j} + \left\| Q_j^k - Q_j^* \right\|_{\mu_j} \right) \left\| Q_j^k - Q_j^* \right\|_{\mu_j}, \quad \text{(by (4a) and Cauchy-Schwarz inequality)} \\
&\quad + \underbrace{\left\langle \mathbb{E}^k \left[Q_{j+1}^k \left(Y_{j+1}^{k+1} (\cdot, \cdot), v_{j+1}^k \left(Y_{j+1}^{k+1} (\cdot, \cdot) \right) \right) - Q_{j+1}^* \left(Y_{j+1}^{k+1} (\cdot, \cdot), v_{j+1}^* \left(Y_{j+1}^{k+1} (\cdot, \cdot) \right) \right) \right], Q_j^k - Q_j^* \right\rangle_{\mu_j}}_{B} \\
&\quad - \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2, \quad \text{(by optimality of } Q^*)
\end{aligned}$$

by noting $Y_{j+1}^{k+1} (\cdot, \cdot) = f_j \left(\cdot, \cdot, W_{j+1}^{k+1} \right)$ in order to reduce expressions. By Cauchy-Schwarz inequality, and then by Jensen inequality:

$$\begin{aligned}
B &\leq \left\| \mathbb{E}^k \left[Q_{j+1}^k \left(Y_{j+1}^{k+1} (\cdot, \cdot), v_{j+1}^k \left(Y_{j+1}^{k+1} (\cdot, \cdot) \right) \right) - Q_{j+1}^* \left(Y_{j+1}^{k+1} (\cdot, \cdot), v_{j+1}^* \left(Y_{j+1}^{k+1} (\cdot, \cdot) \right) \right) \right] \right\|_{\mu_j} \left\| Q_j^k - Q_j^* \right\|_{\mu_j}, \\
&\leq \mathbb{E}^k \left[\left\| Q_{j+1}^k \left(Y_{j+1}^{k+1} (\cdot, \cdot), v_{j+1}^k \left(Y_{j+1}^{k+1} (\cdot, \cdot) \right) \right) - Q_{j+1}^* \left(Y_{j+1}^{k+1} (\cdot, \cdot), v_{j+1}^* \left(Y_{j+1}^{k+1} (\cdot, \cdot) \right) \right) \right\|_{\mu_j} \right] \left\| Q_j^k - Q_j^* \right\|_{\mu_j}, \\
&\leq \left\| Q_{j+1}^k \left(\cdot, v_{j+1}^k (\cdot) \right) - Q_{j+1}^* \left(\cdot, v_{j+1}^* (\cdot) \right) \right\|_{\nu_{j+1}} \left\| Q_j^k - Q_j^* \right\|_{\mu_j}, \quad \text{(by independence of the drawings)} \\
&= \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}} \left\| Q_j^k - Q_j^* \right\|_{\mu_j}.
\end{aligned}$$

So:

$$\begin{aligned}
\mathbb{E}^k [A] \leq b\varepsilon_k &\left(1 + \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}} + \left\| Q_j^k - Q_j^* \right\|_{\mu_j} \right) \left\| Q_j^k - Q_j^* \right\|_{\mu_j} \\
&+ \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}} \left\| Q_j^k - Q_j^* \right\|_{\mu_j} - \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2.
\end{aligned}$$

By developping and using $ab \leq \frac{a^2}{2} + \frac{b^2}{2}$, for any real numbers a, b:

$$\begin{aligned}
\mathbb{E}^k [A] \leq \frac{b\varepsilon_k}{2} &+ \frac{b\varepsilon_k}{2} \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2 + \frac{b\varepsilon_k}{2} \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}}^2 + \frac{b\varepsilon_k}{2} \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2 \\
&+ b\varepsilon_k \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2 - \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2 + \frac{1}{2} \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}}^2 + \frac{1}{2} \left\| Q_j^k - Q_j^* \right\|_{\mu_j}^2.
\end{aligned}$$

Finally:

$$\begin{aligned}
\mathbb{E}^k \left[a_j^{k+1} \right] \leq a_j^k &+ \left(4b\rho_k \varepsilon_k^2 - \rho_k \varepsilon_k \right) a_j^k + b\rho_k \varepsilon_k^2 \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}}^2 \\
&+ b\rho_k \varepsilon_k^2 + \rho_k \varepsilon_k \left\| V_{j+1}^k - V_{j+1}^* \right\|_{\nu_{j+1}}^2 + \rho_k^2 \mathbb{E}^k \left[\left\| \Delta_j^{k+1} K^k \left(X_j^{k+1}, U_j^{k+1}, \cdot, \cdot \right) \right\|_{\mu_j}^2 \right].
\end{aligned}$$

Knowing the assumptions of the theorem and the property (P_{j+1}), one has by Robbins-Siegmund's lemma [RS71]:

$$\lim_{k \to +\infty} a_j^k = l, \text{ and } \sum_{k=1}^{+\infty} \rho_k \varepsilon_k a_j^k < +\infty.$$

As $\sum_{k=1}^{+\infty} \rho_k \varepsilon_k = +\infty$, one has necessarily $\lim_{k\to+\infty} a_j^k = 0$. Moreover, one has:

$$\begin{aligned}\left|V_j^k\left(X_j^{k+1}\right) - V_j^*\left(X_j^{k+1}\right)\right|^2 &\leq \max_{v\in U^{ad}} \left|Q_j^k\left(X_j^{k+1}, v\right) - Q_j^*\left(X_j^{k+1}, v\right)\right|^2, \\ &= \left|Q_j^k\left(X_j^{k+1}, \overline{v}\right) - Q_j^*\left(X_j^{k+1}, \overline{v}\right)\right|^2, \\ &\leq \frac{1}{\pi_{\overline{v}}} \sum_{i=1}^{card(U^{ad})} \pi_{v_i} \left|Q_j^k\left(X_j^{k+1}, v_i\right) - Q_j^*\left(X_j^{k+1}, v_i\right)\right|^2,\end{aligned}$$

Recall that $\pi_{\overline{v}}$ depends on t. If we note $\underline{\pi} = \min_{v\in U^{ad}} \pi_v$, we have that: $\left\|V_j^k - V_j^*\right\|_{L^2}^2 \leq \frac{1}{\underline{\pi}} \left\|Q_j^k - Q_j^*\right\|_{\mu_j}^2$.
By using the dynamics on X and the fact that W_j does not depend on X_{j-1} nor on U_{j-1}, we obtain:

$$\begin{aligned}\left\|V_j^k - V_j^*\right\|_{L^2}^2 &= \mathbb{E}^k\left[\left|V_j^k\left(f_{j-1}\left(X_{j-1}^{k+1}, U_{j-1}^{k+1}, W_j^{k+1}\right)\right) - V_j^*\left(f_{j-1}\left(X_{j-1}^{k+1}, U_{j-1}^{k+1}, W_j^{k+1}\right)\right)\right|^2\right], \\ &= \mathbb{E}^k\left[\mathbb{E}\left[\left|V_j^k\left(f_{j-1}\left(X_{j-1}^{k+1}, U_{j-1}^{k+1}, W_j^{k+1}\right)\right) - V_j^*\left(f_{j-1}\left(X_{j-1}^{k+1}, U_{j-1}^{k+1}, W_j^{k+1}\right)\right)\right|^2 \left|W_j^{k+1}, \mathcal{F}^k\right.\right]\right], \\ &= \mathbb{E}^k\left[\mathbb{E}\left[\sum_{i=1}^{card(U^{ad})} \pi_{v_i} \left|V_j^k\left(f_{j-1}\left(X_{j-1}^{k+1}, v_i, W_j^{k+1}\right)\right) - V_j^*\left(f_{j-1}\left(X_{j-1}^{k+1}, v_i, W_j^{k+1}\right)\right)\right|^2 \left|W_j^{k+1}, \mathcal{F}^k\right.\right]\right], \\ &= \mathbb{E}^k\left[\left\|V_j\left(f_{j-1}\left(\cdot, \cdot, W_j^{k+1}\right)\right) - V_j^*\left(f_{j-1}\left(\cdot, \cdot, W_j^{k+1}\right)\right)\right\|_{\mu_j}^2\right], \\ &= \left\|V_j^k - V_j^*\right\|_{\nu_j}^2.\end{aligned}$$

So: $\left\|V_j^k - V_j^*\right\|_{\nu_j}^2 \leq \frac{1}{\underline{\pi}} \left\|Q_j^k - Q_j^*\right\|_{\mu_j}^2$,
and:

$$\sum_{k=1}^{+\infty} \rho_k \varepsilon_k \left\|V_j^k - V_j^*\right\|_{\nu_j}^2 < +\infty.$$

Hence (P_j) is true. Simultaneously, we showed that: $\lim_{k\to+\infty} a_j^k = 0$.

We have thus that functions $Q_j^k(\cdot,\cdot)$ converge a.s. toward the solution $\left(Q_j^*(\cdot,\cdot)\right)_{j=0,\dots,T}$ of Equation (3). □

Remark 2.3. **On-policy control drawings.** For the efficient use of this algorithm, a question remains open : how to draw policies efficiently ? The only condition we need for convergence is that every possible policy shall be selected infinitely often. In the convergence proof this condition is written as $\underline{\pi} = \min_{v\in U^{ad}} \pi_v > 0$. There are two classical approaches to ensure this, which are often called *on-policy* methods and *off-policy* methods.

Off-policy methods do not take into account the growth of our knowledge of the Q-functions along the iterations. They typically consist in choosing an a priori distribution of the policies to be tested all along the iterations.

On the contrary, on-policy methods aim at selecting more and more policies that seem relevant according to our knowledge of the Q-functions at the current iteration. However, to ensure convergence, we shall still sometimes test policies at random. This is what practitioners call *soft* on-policy control methods.

We choose to test policies in a ε-greedy way, which is an example of soft on-policy method (see [SB98, section 5.4] for more details). In most cases (with probability $1-\varepsilon$), we choose the optimal policy according to our estimate of the Q-functions at the current step, i.e. we choose $u = \arg\min_{v\in\mathbb{U}_j^{ad}} Q_j^k(x, v)$. However, to ensure the convergence of the algorithm, we draw random policies with probability ε.

This technique allows the algorithm to explore the areas where policies seem to be optimal more often.

3. The Mountain Car Task

This problem is explained in [SB98, example 8.2]. Consider the task of driving an underpowered car up a steep mountain road. The difficulty is that gravity is stronger than the car's engine, and even at full throttle the car cannot accelerate up the steep slope. The solution divides in three parts. The point is to build up enough inertia to be able to move up the steep slope to the goal. First, the car has to move in the direction of the goal, and, at a precise point, it has to apply full throttle so that it will climb a little higher up the opposite slope. At this point, the car has enough inertia to carry it up the steep slope even though it is slowing down the whole way. This is a simple example of a continuous control

task where things have to get worse in a sense (farther from the goal) before they can get better. Many control methodologies have great difficulties with tasks of this kind unless explicitly aided by a human designer.

There are three possible actions: full throttle forward (+1), full throttle reverse (−1), and zero throttle (0). The car moves according to a simplified physics. Its position, x_t, and velocity, $\dot{x}_t$, are updated by

$$x_{t+1} = \text{bound}\left[x_t + \dot{x}_{t+1}\right], \tag{5a}$$

$$\dot{x}_{t+1} = \text{bound}\left[\dot{x}_{t+1} + 0.001a_t - 0.0025\cos\left(3x_t\right)\right], \tag{5b}$$

where the bound enforces $-1.2 \leq x_{t+1} \leq 0.5$ and $-0.07 \leq \dot{x}_{t+1} \leq 0.07$. When x_{t+1} reaches the left bound, $\dot{x}_{t+1}$ is reset to zero. When it reaches the right bound, the goal is reached and the episode is terminated. Each episode starts from a random position and velocity uniformly chosen from its feasibility ranges.

To clarify, let us introduce the state variable $s_t = (x_t, \dot{x}_t)$. The problem can thus be written as a minimization problem :

$$\begin{cases} \min\limits_{T\in\mathbb{N},(a_t)_{t\leq T}\in\{-1,0,1\}^{T+1}} & T \\ & \\ & s_{t+1} = f\left(s_t, a_t\right), \\ & s_0 = s, \\ & s_T = S^*, \end{cases}$$

where T denotes the arrival time, S^* denotes the goal area and f denotes the transportation equations (5). Then we introduce the mapping Q defined by:

$$Q\left(s,a\right) = \begin{cases} 1 + \min\limits_{a'} Q\left(f\left(s,a\right),a'\right) & \text{if } s \notin S^*, \\ 0 & \text{if } s \in S^*. \end{cases}$$

Then the update in the algorithm can be summed up as follows:

$$Q^{k+1}\left(\cdot,\cdot\right) = Q^k\left(\cdot,\cdot\right) + \rho^{k+1}\Delta^{k+1}K^{k+1}\left(s^{k+1},a^{k+1},\cdot,\cdot\right),$$

with:

$$\Delta^{k+1} = \begin{cases} \left[1 + \min\limits_{a'} Q^k\left(f\left(s^{k+1},a^{k+1}\right),a'\right)\right] - Q^k\left(s^{k+1},a^{k+1}\right) & \text{if } s^{k+1} \notin S^*, \\ 0 - Q^k\left(s^{k+1},a^{k+1}\right) & \text{if } s^{k+1} \in S^*. \end{cases}$$

The algorithm randomly tries all possible strategies and updates the expected time left to the goal by being at state s^k and applying control a^k.

We draw in figures 2 and 4 the evolution of the position and the optimal control, starting from the bottom of the valley and using the optimal control found by our algorithm after respectively 2000 and 9000 episodes, corresponding to approximately 100 000 and 500 000 iterations. After 2000 episodes, the car needs 108 time steps to reach the goal. After 9000, it needs 101 time steps.

As explained in the problem description, one can observe the complexity of the task by analyzing that the car needs to first reach up the mountain on the right a little, to secondly reach up the left slope sufficiently high, i.e. to gain sufficient inertia, to finally be able, by applying full throttle, to reach up the goal on the right slope.

We draw in figure 5 the Bellman function, which here represents the expected time left to reach the goal as a function of the state (position and speed of the car).

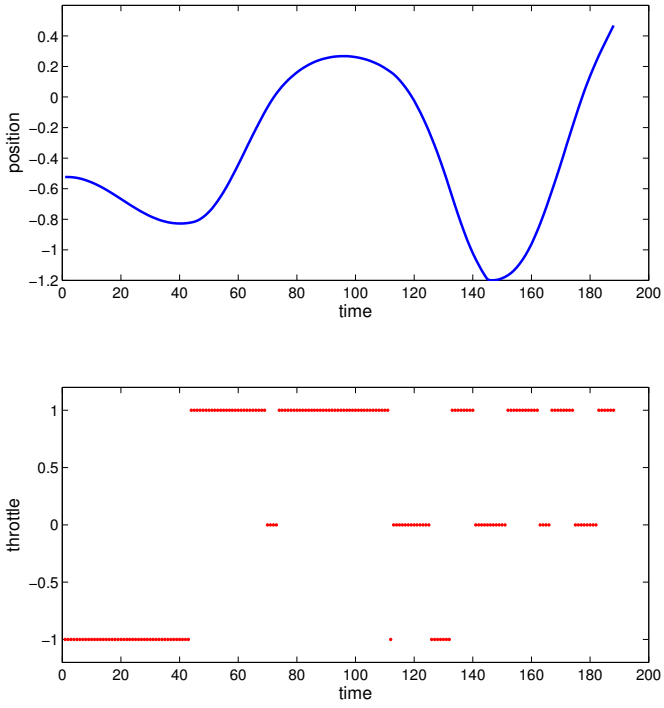

FIGURE 2. Position and throttle of the car, starting from the bottom of the valley and using the greedy policy found by our algorithm after 500 episodes

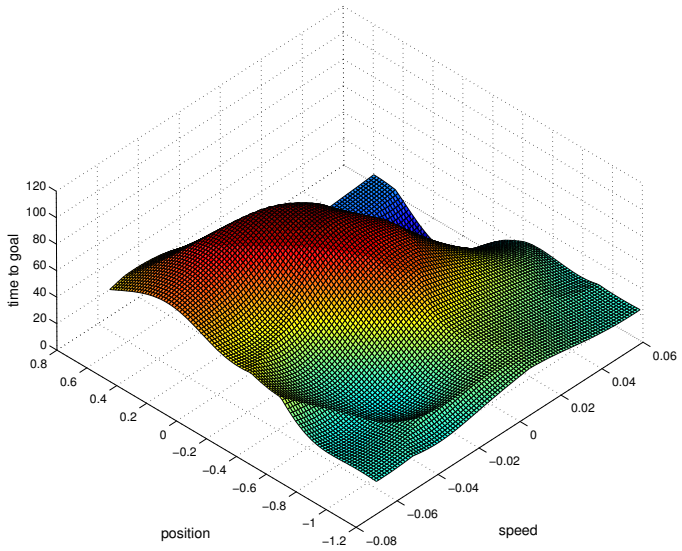

FIGURE 3. Bellman function after 500 episodes

REFERENCES

[Boc55] S. Bochner, *Harmonic analysis and the theory of probability*, University of California Press, Berkeley, 1955.

[JJS94] T. Jaakkola, M.I. Jordan, and S.P. Singh, *Convergence of stochastic iterative dynamic programming algorithms*, Advances in Neural Information Processing Systems (Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, eds.), vol. 6, Morgan Kaufmann Publishers, Inc., 1994, pp. 703–710.

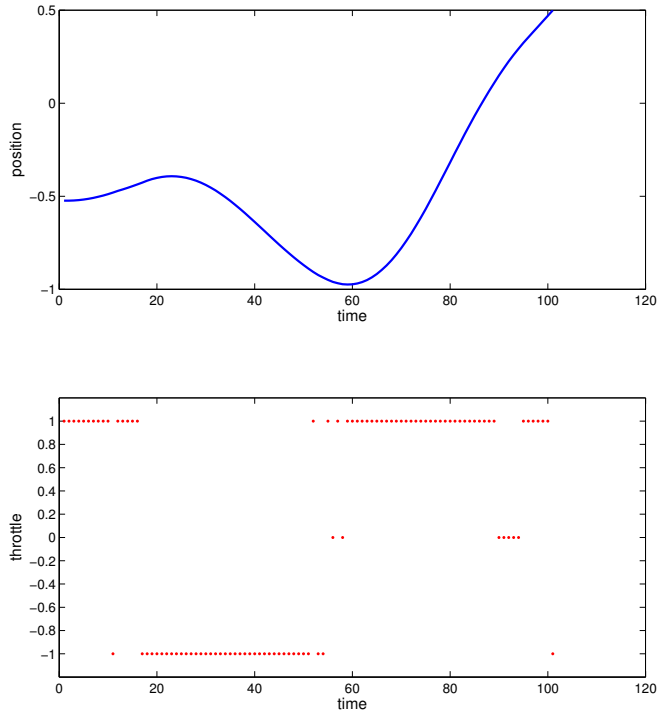

FIGURE 4. Position and throttle of the car, starting from the bottom of the valley and using the greedy policy found by our algorithm after 9000 episodes

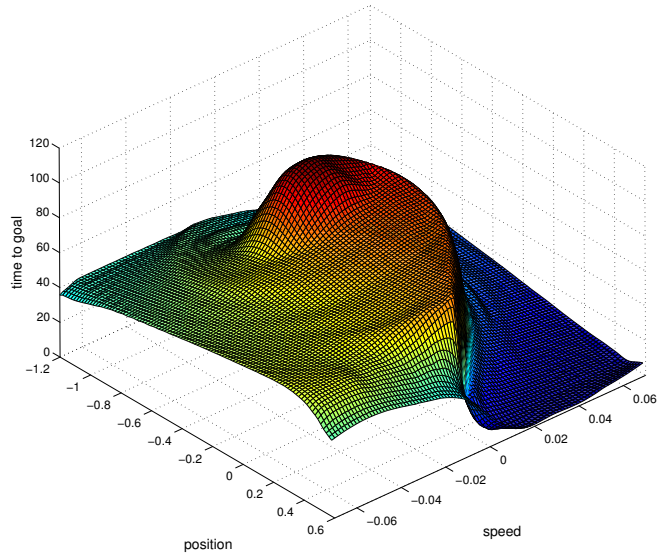

FIGURE 5. Bellman function after 9000 episodes

[Lai03] T.L. Lai, *Stochastic Approximation*, Ann. Stat. **31** (2003), no. 2, 391–406.

[OG00] Dirk Ormoneit and Peter Glynn, *Kernel-based reinforcement learning in average-cost problems: An application to optimal portfolio choice*, NIPS, 2000, pp. 1068–1074.

[OS99] D. Ormoneit and S. Sen, *Kernel-based reinforcement learning*, Tech. Report TR 1999-8, Statistics, Stanford University, 1999.

[RM51] H. Robbins and S. Monro, *A stochastic approximation method*, Annals of Mathematical Statistics **22** (1951), 400–407.

[RS71] H. Robbins and D. Siegmund, *A convergence theorem for nonnegative almost supermartingales and some applications*, Optimizing Methods in Statistics (J.S. Rustagi, ed.), Academic Press, New York, 1971, pp. 233–257.

[SB98] R.S. Sutton and A.G. Barto, *Reinforcement learning, an Introduction*, MIT press Cambridge, 1998.

[Sut88] R.S. Sutton, *Learning to predict by the method of temporal difference*, IEEE Trans. Autom. Control **37** (1988), 332–341.

[Tsi93] J.N. Tsitsiklis, *Asynchronous stochastic approximation and Q-learning*, Machine Learning (1993), no. 16, 185–202.

[Wat89] C. Watkins, *Learning from delayed rewards*, Ph.D. thesis, King's College, Cambridge, 1989.

[WD92] C. Watkins and P. Dayan, *Q-learning*, Machine Learning (1992), no. 8, 279–292.

K. Barty, EDF R&D, 1, avenue du Général de Gaulle, F-92141 Clamart Cedex,
E-mail address: kengy.barty@edf.fr

P. Girardeau, École Nationale Supérieure de Techniques Avancées (ENSTA), also with EDF R&D
E-mail address: pierre.girardeau@ensta.fr

J.-S. Roy, EDF R&D, 1, avenue du Général de Gaulle, F-92141 Clamart Cedex,
E-mail address: jean-sebastien.roy@edf.fr

C. Strugarek, EDF R&D, also with the École Nationale Supérieure de Techniques Avancées (ENSTA) and the École Nationale des Ponts et Chaussées (ENPC)
E-mail address: cyrille.strugarek@edf.fr