# Optimal Tuning of Continual Online Exploration in Reinforcement Learning

Youssef Achbany, Francois Fouss, Luh Yen, Alain Pirotte, and Marco Saerens

Information Systems Research Unit (ISYS)
Place des Doyens 1, Université de Louvain, Belgium
{youssef.achbany, francois.fouss, luh.yen, alain.pirotte,
marco.saerens}@ucLouvain.be

**Abstract.** This paper presents a framework allowing to tune continual exploration in an optimal way. It first quantifies the rate of exploration by defining the **degree of exploration** of a state as the probability-distribution entropy for choosing an admissible action. Then, the exploration/exploitation tradeoff is stated as a **global optimization problem**: find the exploration strategy that minimizes the expected cumulated cost, while maintaining fixed degrees of exploration at same nodes. In other words, "exploitation" is maximized for constant "exploration". This formulation leads to a set of nonlinear updating rules reminiscent of the value-iteration algorithm. Convergence of these rules to a local minimum can be proved for a stationary environment. Interestingly, in the deterministic case, when there is no exploration, these equations reduce to the Bellman equations for finding the shortest path while, when it is maximum, a full "blind" exploration is performed.

## 1 Introduction

One of the specific challenges of reinforcement learning is the tradeoff between exploration and exploitation. Exploration aims to continually try new ways of solving the problem, while exploitation aims to capitalize on already well-established solutions. Exploration is especially relevant when the environment is changing, i.e. nonstationary. In this case, good solutions can deteriorate over time and better solutions can appear. Without exploration, the system sends agents only along the up-to-now best path without exploring alternative paths. The system is therefore unaware of the changes and its performance inevitably deteriorates with time. One of the key features of reinforcement learning is that it explicitly addresses the exploration/exploitation issue as well as the online estimation of the probability distributions in an integrated way [18].

This work makes a clear distinction between "*preliminary*" or "*initial exploration*", and "*continual online exploration*". The objective of *preliminary exploration* is to discover relevant goals, or destination states, and to estimate a first optimal policy for exploiting them. On the other hand, *continual online exploration* aims to continually explore the environment, after the preliminary exploration stage, in order to adjust the policy to changes in the environment.

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 790–800, 2006.

In the case of *preliminary exploration*, two further distinctions are often made [19,20,21,22]. A first group of strategies uses randomness for exploration and is often referred to as *undirected exploration*. Control actions are selected with a probability distribution, taking the expected cost into account. The second group, referred to as *directed exploration*, uses domain-specific knowledge for guiding exploration [19,20,21,22]. Usually, *directed exploration* provides better results in terms of learning time and cost.

On the other hand, "*continual online exploration*" can be performed by, for instance, re-exploring the environment either periodically or continually [6,15] by using a $\epsilon$-greedy or a Boltzmann exploration strategy. For instance, joint estimation of the exploration strategy and the state-transition probabilities for continual online exploration can be performed within the SARSA framework [14,16,18].

This work presents a unified framework integrating exploitation and exploration for *undirected, continual, exploration*. Exploration is formally defined as the association of a probability distribution to the set of admissible control actions in each state (choice randomization). The *rate of exploration* is quantified with the concept of **degree of exploration**, defined as the (Shannon) entropy [10] of the probability distribution for the set of admissible actions in a given state. If no exploration is performed, the agents are routed on the best path with probability one – they just exploit the solution. With exploration, the agents continually explore a possibly changing environment to keep current with it. When the entropy is zero in a state, no exploration is performed from this state, while, when the entropy is maximal, a full, blind exploration with equal probability of choosing any action is performed.

The online exploration/exploitation issue is then stated as a **global optimization problem**: learn the exploration strategy that minimizes the expected cumulated cost from the initial state to the goal while maintaining a fixed degree of exploration. In other words, "exploitation" is maximized for constant "exploration". This problem leads to a set of nonlinear equations defining the optimal solution. These equations can be solved by iterating them until convergence, which is proved for a stationary environment and a particular initialization strategy. They provide the action policy (the probability distribution of choosing an action in a given state) that minimizes the average cost from the initial state to the destination states, given the degree of exploration in each state. Interestingly, when the degree of exploration is zero in all states, which corresponds to the deterministic case, the nonlinear equations reduce to the Bellman equations for finding the shortest path from the initial state to the destination states. The main drawback of this method is that it is computationally demanding since it relies on iterative algorithms like the value-iteration algorithm.

For the sake of simplicity, we first concentrate here on "*deterministic shortest-path problem*", as defined for instance in [5], where any chosen control action deterministically drives the agent to a unique successor state. On the other hand, if the actions have uncertain effects, the resulting state is given by a probability distribution and one speaks of "*stochastic shortest-path problems*".

In this case, a probability distribution on the successor states is introduced and it must be estimated by the agents; stochastic shortest-path problems are studied in Section 4.

Section 2 introduces the notations, the standard deterministic shortest-path problem, and the management of continual exploration. Section 3 describes our procedure for solving the deterministic shortest-path problem with continual exploration, while the stochastic shortest-path problem is discussed in Section 4. Section 5 is the conclusion.

## 2 Statement of the Problem and Notations

### 2.1 Statement of the Problem

During every state transition, a finite cost $c(k, u)$ is incurred when leaving state $k \in \{1, 2, \ldots, n\}$ while executing a control action $u$ selected from a set $U(k)$ of admissible actions, or choices, available in state $k$. The cost can be positive (penalty), negative (reward), or zero provided that no cycle exists whose total cost is negative. This is a standard requirement in shortest-path problems [8]; indeed, if such a cycle exists, then traversing it an arbitrary large number of times would result in a path with an arbitrary small cost so that a best path could not be defined. In particular, this implies that, if the graph of the states is nondirected, all costs are nonnegative.

The **control action** $u$ is chosen according to a **policy** $\Pi$ that maps every state $k$ to the set $U(k)$ of admissible actions with a certain probability distribution, $\pi_k(u)$, with $u \in U(k)$. Thus the policy associates to each state $k$ a probability distribution on the set of admissible actions $U(k)$: $\Pi \equiv \{\pi_k(u), k = 1, 2, \ldots, n\}$. For instance, if the admissible actions in state $k$ are $U(k) = \{u_1, u_2, u_3\}$, the distribution $\pi_k(u)$ specifies three probabilities $\pi_k(u_1)$, $\pi_k(u_2)$, and $\pi_k(u_3)$. The **degree of exploration** is quantified as the entropy of this probability distribution (see next section). Randomized choices are common in a variety of fields, for instance decision sciences [13] or game theory, where they are called mixed strategies (see, e.g., [12]). Thus, the problem tackled in this section corresponds to a **randomized shortest-path problem**.

Moreover, we assume that once the action has been chosen, the next state $k'$ is known deterministically, $k' = f_k(u)$ where $f$ is a one-to-one mapping between (states, actions) and resulting state. We assume that different actions lead to different states. This framework corresponds to a deterministic shortest-path problem. A simple modeling of this problem would do without actions and directly defined state-transition probabilities. The more general formalism fits full stochastic problems for which both the choice of actions and the state transitions are governed by probability distributions (see Section 4).

We assume, as in [5], that there is a special cost-free **destination** or **goal state**; once the system has reached that state, it remains there at no further cost. The goal is to minimize the **total expected cost** $V_{\Pi}(k_0)$ (Equation (2.1))

accumulated over a path $k_0, k_1, ...$ in the graph starting from an initial (or source) state $k_0$:

$$V_{\Pi}(k_0) = E_{\Pi}\left[\sum_{i=0}^{\infty} c(k_i, u_i)\right] \tag{2.1}$$

The expectation $E_{\Pi}$ is taken on the policy $\Pi$ that is, on all the random choices of action $u_i$ in state $k_i$.

Moreover, we consider a problem structure such that termination is guaranteed, at least under an optimal policy. Thus, the horizon is finite, but its length is random and it depends on the policy. The conditions for which termination holds are equivalent to establishing that the destination state can be reached in a finite number of steps from any potential initial state; for a rigorous treatment, see [3,5].

### 2.2 Controling Exploration by Defining Entropy at Each State

The **degree of exploration** $E_k$ at each state $k$ is defined by

$$E_k = -\sum_{i \in U(k)} \pi_k(i) \log \pi_k(i) \tag{2.2}$$

which is simply the entropy of the probability distribution of the control actions in state $k$ [9,10]. $E_k$ characterizes the uncertainty about the choice at state $k$. It is equal to zero when there is no uncertainty at all ($\pi_k(i)$ reduces to a Kronecker delta); it is equal to $\log(n_k)$, where $n_k$ is the number of admissible choices at node $k$, in the case of maximum uncertainty, $\pi_k(i) = 1/n_k$ (a uniform distribution).

The **exploration rate** $E_k^r = E_k / \log(n_k)$ is the ratio between the actual value of $E_k$ and its maximum value. It takes its values in the interval $[0, 1]$. Fixing the entropy at a state sets the exploration level out of this state; increasing the entropy increases exploration up to the maximal value, in which case there is no more exploitation since the next action is chosen completely at random, with a uniform distribution, without taking the costs into account. This way, the agents can easily control their exploration by adjusting the exploration rates.

## 3 Optimal Policy Under Exploration Constraints for Deterministic Shortest-Path Problems

### 3.1 Optimal Policy and Expected Cost

We turn to the determination of the optimal policy under exploration constraints. More precisely, we will seek the policy $\Pi \equiv \{\pi_k(u), k = 1, 2, \ldots, n\}$, for which the expected cost $V_{\Pi}(k_0)$ from initial state $k_0$ is minimal while maintaining a given degree of exploration at each state $k$. The destination state is an absorbing state, i.e., with no outgoing link. Computing the expected cost (2.1) from any state $k$ is similar to computing the average first-passage time in the associated Markov chain [11]. The problem is thus to find the transition

probabilities leading to the minimal expected cost, $V^*(k_0) = \min_{\Pi}(V_{\Pi}(k_0))$. It can be formulated as a constrained optimization problem involving a Lagrange function.

In [1], we derive the optimal probability distribution of control actions in state $k$, which is a logit distribution:

$$\pi_k^*(i) = \frac{\exp\left[-\theta_k \left(c(k,i) + V^*(k_i')\right)\right]}{\sum_{j \in U(k)} \exp\left[-\theta_k \left(c(k,j) + V^*(k_j')\right)\right]}, \tag{3.1}$$

where $k_i' = f_k(i)$ is a following state and $V^*$ is the optimal (minimum) expected cost given by

$$\begin{cases} V^*(k) = \sum_{i \in U(k)} \pi_k^*(i)\left[c(k,i) + V^*(k_i')\right], \text{ with } k_i' = f_k(i) \text{ and } k \neq d \\ V^*(d) = 0, \text{ for the destination state } d \end{cases} \tag{3.2}$$

The control actions probability distribution (3.1) is often called "Boltzmann distributed exploration". In Equation (3.1), $\theta_k$ must be chosen in order to satisfy

$$\sum_{i \in U(k)} \pi_k(i) \log \pi_k(i) = -E_k \tag{3.3}$$

for each state $k$ and given $E_k$. It takes its values in $[0, \infty]$. Of course if, for some state, the number of possible control actions reduces to one (no choice), no entropy constraint is introduced. Since Equation (3.3) has no analytical solution, $\theta_k$ must be computed numerically in terms of $E_k$. This is in fact quite easy since it can be shown that the function $\theta_k(E_k)$ is strictly monotonic decreasing, so that a line search algorithm (such as the bisection method, see [2]) or a simple binary search can efficiently find the $\theta_k$ value corresponding to a given $E_k$ value.

Equation (3.1) has a simple appealing interpretation: choose preferably (with highest probability) action $i$ leading to state $k_i'$ of lowest expected cost, including the cost of performing the action, $c(k,i) + V^*(k_i')$. Thus, the agent is routed preferably to the state which is nearest (on average) to the destination state.

The same necessary optimality conditions can also be expressed in terms of the $Q$-values coming from the popular $Q$-learning framework [18,23,24]. Indeed, in the deterministic case, the $Q$-value represents the expected cost from state $k$ when choosing action $i$, $Q(k,i) = c(k,i) + V(k_i')$. The relationship between $Q$ and $V$ is thus simply $V(k) = \sum_{i \in U(k)} \pi_k(i)\, Q(k,i)$; we thus easily obtain

$$\begin{cases} Q^*(k,i) = c(k,i) + \sum_{i \in U(k_i')} \pi_{k_i'}^*(i)\, Q^*(k_i', i), \text{ with } k_i' = f_k(i) \text{ and } k \neq d \\ Q^*(d,i) = 0, \text{ for the destination state } d \end{cases} \tag{3.4}$$

and the $\pi_k^*(i)$ are given by

$$\pi_k^*(i) = \frac{\exp\left[-\theta_k Q^*(k,i)\right]}{\sum_{j \in U(k)} \exp\left[-\theta_k Q^*(k,j)\right]} \tag{3.5}$$

which corresponds to a Boltzmann exploration involving the $Q$-value. Thus, a Boltzmann exploration involving the $Q$-value may be considered as "optimal" since it provides the best expected performances for fixed degrees of exploration.

### 3.2 Computation of the Optimal Policy

Equations (3.1) and (3.2) suggest an iterative procedure very similar to the well-known value-iteration algorithm for the computation of both the expected cost and the policy.

More concretely, we consider that agents are sent from the initial state and that they choose an action $i$ in each state $k$ with probability distribution $\pi_k(u = i)$. The agent then performs the chosen action, say action $i$, and incurs the associated cost, $c(k, i)$ (which, in a non-stationary environment, may vary over time), together with the new state, $k'$. This allows the agent to update the estimates of the cost, of the policy, and of the average cost until destination; these estimates will be denoted by $\widehat{c}(k, i)$, $\widehat{\pi}_k(i)$ and $\widehat{V}(k)$ and are known (shared) by all the agents.

1. **Initialization phase**
   - Choose an initial policy, $\widehat{\pi}_k(i)$, $\forall i, k$, satisfying the exploration rate constraints (3.3) and
   - Compute the corresponding expected cost until destination $\widehat{V}(k)$ by using any procedure for solving the set of linear equations (3.2) where we substitue $V^*(k)$, $\pi_k^*(i)$ by $\widehat{V}(k)$, $\widehat{\pi}_k(i)$. The $\widehat{\pi}_k(i)$ are kept fixed in the initialization phase. Any standard iterative procedure (for instance, a Gauss-Seidel like algorithm) for computing the expected cost until absorption in a Markov chain could be used (see [11]).
2. **Computation of the policy and the expected cost under exploration constraints**
   For each visited state $k$, do until convergence:
   - Choose an action $i$ with current probability estimate $\widehat{\pi}_k(i)$, observe the current cost $c(k, i)$ for performing this action, update its estimate $\widehat{c}(k, i)$, and jump to the next state, $k'_i$

$$\widehat{c}(k, i) \leftarrow c(k, i) \tag{3.6}$$

   - Update the probability distribution for state $k$ as:

$$\widehat{\pi}_k(i) \leftarrow \frac{\exp\left[-\widehat{\theta}_k \left(\widehat{c}(k, i) + \widehat{V}(k'_i)\right)\right]}{\sum_{j \in U(k)} \exp\left[-\widehat{\theta}_k \left(\widehat{c}(k, j) + \widehat{V}(k'_j)\right)\right]}, \tag{3.7}$$

   where $k'_i = f_k(i)$ and $\widehat{\theta}_k$ is set in order to respect the given degree of entropy (see Equation (3.3)).

– Update the expected cost of state $k$:

$$\begin{cases} \widehat{V}(k) \leftarrow \sum\limits_{i \in U(k)} \widehat{\pi}_k(i) \left[\widehat{c}(k,i) + \widehat{V}(k_i')\right], \text{ with } k_i' = f_k(i) \text{ and } k \neq d \\ \widehat{V}(d) \leftarrow 0, \text{ where } d \text{ is the destination state} \end{cases} \tag{3.8}$$

The convergence of these updating equations is proved for a stationary environment in [1]. However, the described procedure is computationally demanding since it relies on iterative procedures like the value-iteration algorithm in Markov decision processes.

Thus, the above procedure allows to optimize the expected cost $V(k_0)$ and to obtain a local minimum of this criterion. It does not guarantee to converge to a global minimum, however. Whether $V(k_0)$ has only one global minimum or many local minima remains an open question.

Notice also that, while the initialization phase is necessary in our convergence proof, other simpler initialization schemes could also be applied. For instance, set initially $\widehat{c}(k,i) = 0$, $\widehat{\pi}_k(i) = 1/n_k$, $\widehat{V}(k) = 0$, where $n_k$ is the number of admissible actions in state $k$; then proceed by directly applying updating rules (3.7) and (3.8). While convergence is not proved in this case, we observed that this updating rule works well in practice; in particular, we did not observe any convergence problem. This rule is used in the experiments presented in [1].

### 3.3 Some Limit Cases

We will now show that when the degree of exploration is zero for all states, the nonlinear equations reduce to Bellman's equations for finding the shortest path from the initial state to the destination state.

Indeed, from Equations (3.7)-(3.8), if the parameter $\widehat{\theta}_k$ is very large, which corresponds to a near-zero entropy, the probability of choosing the action with the lowest value of $(\widehat{c}(k,i) + \widehat{V}(k_i'))$ dominates all the others. In other words, $\widehat{\pi}_k(j) \simeq 1$ for the action $j$ corresponding to the lowest average cost (including the action cost), while $\widehat{\pi}_k(i) \simeq 0$ for the other alternatives $i \neq j$. Equations (3.8) can therefore be rewritten as

$$\begin{cases} \widehat{V}(k) \leftarrow \min\limits_{i \in U(k)} \left[\widehat{c}(k,i) + \widehat{V}(k_i')\right], \text{ with } k_i' = f_k(i) \text{ and } k \neq d \\ \widehat{V}(d) \leftarrow 0, \text{ where } d \text{ is the destination state} \end{cases} \tag{3.9}$$

which are Bellman's updating equations for finding the shortest path to the destination state [4,5]. In terms of $Q$-values, the optimality conditions reduce to

$$\begin{cases} Q^*(k,i) = c(k,i) + \min\limits_{i \in U(k)} Q^*(k_i', i), \text{ with } k_i' = f_k(i) \text{ and } k \neq d \\ Q^*(d,i) = 0, \text{ for the destination state } d \end{cases} \tag{3.10}$$

On the other hand, when $\widehat{\theta}_k = 0$, the choice probability distribution reduces to $\widehat{\pi}_k(i) = 1/n_k$, and the degree of exploration is maximum for all states. In this

case, the nonlinear equations reduce to the linear equations allowing to compute the average cost for reaching the destination state from the initial state in a Markov chain with transition probabilities equal to $1/n_k$. In other words, we then perform a "blind" random exploration, for the choice probability distribution.

Any intermediary setting $0 < E_k < \log(n_k)$ leads to an optimal exploration vs. exploitation strategy minimizing the expected cost, and favoring short paths to the solution. In [1], we further show that, if the graph of states is directed and acyclic, the nonlinear equations can easily be solved by performing a single backward pass from the destination state.

Experimental simulations illustrating the behaviour of the algorithm, as well as comparisons with a naive Boltzmann and a $\epsilon$-greedy exploration strategy, are provided in [1].

## 4 Optimal Policy Under Exploration Constraints for Stochastic Shortest Path Problems

We now consider **stochastic shortest path problems** for which, once an action has been performed, the transition to the next state is no longer deterministic but stochastic [5]. More precisely, when an agent chooses action $i$ in state $k$, it jumps to state $k'$ with a probability $\mathrm{P}(s = k'|u = i, s = k) = p_{kk'}(i)$ (transition probabilities). Notice that there are now two different probability distributions associated to the system: the probability of choosing an action $i$ within the state $k$, $\pi_k(i)$, and the probability of jumping to a state $s = k'$ after having chosen the action $i$ within the state $k$, $p_{kk'}(i)$.

By first-step analysis (see [1]), the recurence relations allowing to compute the expected cost $V_{\Pi}(k)$, given policy $\Pi$ are easily found:

$$\begin{cases} V_{\Pi}(k) = \sum\limits_{i \in U(k)} \pi_k(i) \left[ c(k,i) + \sum\limits_{k'=1}^{n} p_{kk'}(i)\, V_{\Pi}(k') \right], \\ V_{\Pi}(d) = 0, \text{ where } d \text{ is the destination state} \end{cases} \tag{4.1}$$

Furthermore, by defining the average cost when having chosen control action $i$ in state $k$ by $\overline{V}_{\Pi}(k,i) = \sum_{k'} p_{kk'}(i) V_{\Pi}(k')$, Equation (4.1) can be rewritten as

$$\begin{cases} V_{\Pi}(k) = \sum\limits_{i \in U(k)} \pi_k(i) \left[ c(k,i) + \overline{V}_{\Pi}(k,i) \right], \\ V_{\Pi}(d) = 0, \text{ where } d \text{ is the destination state} \end{cases} \tag{4.2}$$

Thus, the optimal policy is obtained by substituting $V_{\Pi}(k'_i)$ by $\overline{V}^*(k,i)$ in both (3.1) and (3.2):

$$\pi^*_k(i) = \frac{\exp\left[-\theta_k \left( c(k,i) + \overline{V}^*(k,i) \right)\right]}{\sum\limits_{j \in U(k)} \exp\left[-\theta_k \left( c(k,j) + \overline{V}^*(k,j) \right)\right]} \tag{4.3}$$

The details are provided in [1]. The additional difficulty here, in comparison with a deterministic problem, is that the probability distributions $p_{kk'}(i)$, if unknown, have to be estimated on-line, together with the costs and the distribution of the randomized control actions [18].

### 4.1 On-Line Estimation of the Transition Probabilities

The transition probabilities $p_{kk'}(i)$ might be unknown and, consequently, should be estimated on-line, together with the costs and the distribution of the randomized control actions [18]. An alternative solution is to directly estimate the average cost $\overline{V}_{\Pi}(k,i) = \sum_{k'} p_{kk'}(i) V_{\Pi}(k')$ based on the observation of the value of $V_{\Pi}$ in the next state $k'$. There is a large range of potential techniques for solving this issue, depending on the problem at hand (see for example [7]). One could simply use an exponential smoothing, leading to $\widehat{\overline{V}}(k,i) \leftarrow \alpha \widehat{V}(k') + (1-\alpha)\widehat{\overline{V}}(k,i)$, or a stochastic approximation scheme, $\widehat{\overline{V}}(k,i) \leftarrow \widehat{\overline{V}}(k,i) + \alpha \left[\widehat{V}(k') - \widehat{\overline{V}}(k,i)\right]$, which converges for a suitable decreasing policy of $\alpha$ [17].

This leads to the following updating rules:

For each visited state $k$, do until convergence:

- Choose an action $i$ with current probability estimate $\widehat{\pi}_k(i)$, observe the current cost $c(k,i)$ for performing this action, update its estimate $\widehat{c}(k,i)$ by

$$\widehat{c}(k,i) \leftarrow c(k,i) \tag{4.4}$$

- Perform the action $i$ and observe the current value $\widehat{V}(k')$ of the next state $k'$. Update $\widehat{\overline{V}}(k,i)$ accordingly (here, we choose the exponential smoothing scheme),

$$\widehat{\overline{V}}(k,i) \leftarrow \alpha \overline{V}(k') + (1-\alpha)\widehat{\overline{V}}(k,i) \tag{4.5}$$

- Update the probability distribution for state $k$ as:

$$\widehat{\pi}_k(i) \leftarrow \frac{\exp\left[-\widehat{\theta}_k \left(\widehat{c}(k,i) + \widehat{\overline{V}}(k,i)\right)\right]}{\sum\limits_{j \in U(k)} \exp\left[-\widehat{\theta}_k \left(\widehat{c}(k,j) + \widehat{\overline{V}}(k,j)\right)\right]}, \tag{4.6}$$

  where $\widehat{\theta}_k$ is set in order to respect the prescribed degree of entropy (see Equation (3.3)).
- Update the expected cost of state $k$ asynchronously:

$$\begin{cases} \widehat{V}(k) = \sum\limits_{i \in U(k)} \pi_k(i) \left[\widehat{c}(k,i) + \widehat{\overline{V}}(k,i)\right], \\ \widehat{V}(d) = 0, \text{ where } d \text{ is the destination state} \end{cases} \tag{4.7}$$

This iterative scheme is closely linked to the SARSA on-policy control algorithm [14,16,18]; a discussion of these relationships is provided in [1].

## 5 Conclusions

We have presented a model integrating continual exploration and exploitation in a common framework. The exploration rate is controlled by the entropy of the choice probability distribution defined on the states of the system. When no exploration is performed (zero entropy on each node), the model reduces to the common value-iteration algorithm computing the minimum cost policy. On the other hand, when full exploration is performed (maximum entropy on each node), the model reduces to a "blind" exploration, without considering the costs. The main drawback of the present approach is that it is computationally demanding since it relies on iterative procedures such as the value-iteration algorithm.

Further work will investigate the relationships with SARSA, as well as alternative cost formulations, such as the "average cost per step". We also plan to exploit the proposed exploration framework in Markov games.

## Acknowledgments

Part of this work has been funded by projects with the "Région wallonne" and the "Région de Bruxelles-Capitale".

## References

1. Y. Achbany, F. Fouss, L. Yen, A. Pirotte, and M. Saerens. Tuning continual exploration in reinforcement learning. *Technical report, http://www.isys.ucl.ac.be/staff/francois/Articles/Achbany2005a.pdf*, 2005.
2. M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: Theory and algorithms*. John Wiley and Sons, 1993.
3. D. P. Bertsekas. *Neuro-dynamic programming*. Athena Scientific, 1996.
4. D. P. Bertsekas. *Network optimization: continuous and discrete models*. Athena Scientific, 1998.
5. D. P. Bertsekas. *Dynamic programming and optimal control*. Athena sientific, 2000.
6. J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in Neural Information Processing Systems 6 (NIPS6)*, pages 671–678, 1994.
7. R. G. Brown. *Smoothing, forecasting and prediction of discrete time series*. Prentice-hall, 1962.
8. N. Christofides. *Graph theory: An algorithmic approach*. Academic Press, 1975.
9. T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley and Sons, 1991.
10. J. N. Kapur and H. K. Kesavan. *Entropy optimization principles with applications*. Academic Press, 1992.
11. J. G. Kemeny and J. L. Snell. *Finite markov chains*. Springer-Verlag, 1976.
12. M. J. Osborne. *An introduction to game theory*. Oxford University Press, 2004.
13. H. Raiffa. *Decision analysis*. Addison-Wesley, 1970.
14. G. Rummery and M. Niranjan. On-line q-learning using connectionist systems. *Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Departement*, 1994.

15. G. Shani, R. Brafman, and S. Shimony. Adaptation for changing stochastic environments through online pomdp policy learning. In *Workshop on Reinforcement Learning in Non-Stationary Environments , ECML 2005*, pages 61–70, 2005.
16. S. Singh and R. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158, 1996.
17. J. C. Spall. *Introduction to stochastic search and optimization*. Wiley, 2003.
18. R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. The MIT Press, 1998.
19. S. Thrun. Efficient exploration in reinforcement learning. *Technical report, School of Computer Science, Carnegie Mellon University*, 1992.
20. S. Thrun. The role of exploration in learning control. In D. White and D. Sofge, editors, *Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold, Florence, Kentucky 41022, 1992.
21. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
22. K. Verbeeck. Coordinated exploration in multi-agent reinforcement learning. PhD thesis, Vrije Universiteit Brussel, Belgium, 2004.
23. J. C. Watkins. Learning from delayed rewards. PhD thesis, King's College of Cambridge, UK, 1989.
24. J. C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.