Optimistic Planning for Belief-Augmented Markov Decision Processes

Raphael Fonteneau*†, Lucian Buşoniu‡, Rémi Munos†
*Department of Electrical Engineering and Computer Science, University of Liège, BELGIUM
†SequeL Team, Inria Lille - Nord Europe, FRANCE
Email: {raphael.fonteneau, remi.munos}@inria.fr
‡Université de Lorraine, CRAN, UMR 7039 and CNRS, CRAN, UMR 7039, FRANCE
Email: lucian@busoniu.net

***Abstract*—This paper presents the Bayesian Optimistic Planning (BOP) algorithm, a novel model-based Bayesian reinforcement learning approach. BOP extends the planning approach of the Optimistic Planning for Markov Decision Processes (OP-MDP) algorithm [10], [9] to contexts where the transition model of the MDP is initially unknown and progressively learned through interactions within the environment. The knowledge about the unknown MDP is represented with a probability distribution over all possible transition models using Dirichlet distributions, and the BOP algorithm plans in the belief-augmented state space constructed by concatenating the original state vector with the current posterior distribution over transition models. We show that BOP becomes Bayesian optimal when the budget parameter increases to infinity. Preliminary empirical validations show promising performance.**

I. Introduction

Learning algorithms for planning and decision making have become increasingly popular in the past few years, and they have attracted researchers among several types of applications such as financial engineering [23], medicine [29], robotics [31], [33], and many sub-domains of artificial intelligence [38]. By collecting data about the underlying environment, such algorithms have the ability to learn how to behave near-optimally with respect to a given optimality criterion.

Several challenges need to be addressed when designing such algorithms. In particular, one of the main difficulties is to solve the so-called Exploration versus Exploitation (E/E) dilemma: at a given time-step of the process, the algorithm must both (i) take a decision which is of good quality regarding information that has been collected so far (the *exploitation* part) and (ii) open the door for collecting new information about the (unknown) underlying environment in order to take better decisions in the future (the *exploration* part). Such a problem has been intriguing researchers for many decades: in the sixties, the optimal control community was already developing the dual control theory [18] (“dual” referring to the dual objective E/E), proving that such a dilemma should theoretically be solvable using Dynamic Programming [5].

In the end of the eighties, the popularization of Reinforcement Learning (RL) [37] gave a new impulse to the research community working on the design of efficient algorithms for learning how to plan in unknown environments, and the E/E dilemma was re-discovered in the light of the RL paradigm. As a first step, heuristic-type of solutions were proposed ($\epsilon-$greedy policies, Boltzmann exploration), but later in the end of the nineties, new horizons were opened thanks to techniques coming from Bayesian statistics, leading to Bayesian RL (BRL) [14], [36]. The main asset of BRL was to formalize in an elegant manner the E/E dilemma so that one could theoretically solve it. However, in practice, BRL approaches revealed themselves to be almost intractable, except in the case of $k-$armed bandit problems where the Bayesian approach leads to the well known *Gittins indices* [20]. Despite computational challenges, BRL has become more and more popular in the last decade [32], even if standard BRL algorithms were still outperformed by classic RL algorithms (see for instance [6]).

More recently, a new generation of algorithms based on tree search techniques has lead to a huge breakthrough in RL in terms of empirical performance. In particular, Monte Carlo Tree Search (MCTS) techniques [13], [28], and in particular the UCT algorithm (for “Upper Confidence Trees”, see [25]) have allowed to tackle large scale problems such as the game of Go [19]. Such techniques are actually being exported to the BRL field of research, leading to new efficient algorithms [34], [3], [21].

The contribution detailed in this paper stands within this context, in between model-based BRL and tree search algorithms. We present the BOP algorithm (for “Bayesian Optimistic Planning”), a novel model-based BRL algorithm. BOP extends the principle of the OP-MDP algorithm (for “Optimistic Planning for MDPs”, see [10], [9]) to the case were the model of the environment is initially unknown and needs to be learned through interactions. The optimistic approach for planning proposed in the OP-MDP algorithm is derived in a BA-MDP (for “Belief-Augmented MDP”, see [16]) obtained by concatenating the actual state with a posterior distribution over possible transition models. The algorithm builds a belief-augmented planning tree by taking the current BA-state at the root node ; it iteratively expands new nodes by adding to them, for all possible actions, all subsequent BA-states. Since BOP is designed to be used on-line, the number of expansions is fixed to a given budget parameter n in order to limit the computation time. An optimistic planning procedure is used to allocate efficiently this budget by expanding the most promising BA-states first. Such an approach is made tractable by assuming one independent Dirichlet distribution for each state-action pair, which allows to constrain the branching factor of the exploration trees. This branching factor turns out to be the same as in the OP-MDP framework. Like OP-MDP, BOP can be reinterpreted as a branch-and-bound-type optimization technique in a space of tree-policies, and the analysis of OP-MDP also applies, showing that BOP leads to Bayesian-optimal decisions as the budget parameter n converges towards infinity. The approach is illustrated on the standard 5-state chain MDP [36].

The remainder of this paper is organized as follows: in Section II, we discuss some related work using the optimistic principle in the context of MDPs. Section III formalizes the model-based BRL problem considered in this paper. Section IV presents the main contribution of this paper, the BOP algorithm. In Section V, BOP is reinterpreted as a branch-and-bound-type optimization technique, and its convergence towards Bayesian optimality is stated in Section VI. Section VII presents some simulation results and Section VIII concludes.

II. Related optimistic approaches

The *optimism in the face of uncertainty* paradigm has already lead to several successful results (see [28] for a extensive view of the use of the optimistic principles applied to planning and optimization). Optimism has been specifically used in the following contexts: (i) multi-armed bandit problems (which can be seen as 1-state MDPs) [4], [8], (ii) planning algorithms for deterministic systems [22] and stochastic systems [25], [39], [7], [3], [10], [9], [40] when the system dynamics / transition model is known, and also (iii) optimization of unknown functions only accessible through sampling [27].

The optimistic principle has also been used for addressing the E/E dilemma for MDPs when the transition model is unknown and progressively learned through interactions with the environment. For instance, the R-MAX algorithm [6] assumes optimistic rewards for less visited transitions. The UCRL / UCRL2 algorithms [30], [24] also adopt an optimistic approach to face the E/E dilemma using upper confidence bounds. Very recently, [12] proposed to solve the E/E dilemma in a context where one can sample MDPs from a known (computational) distribution, which has the flavor of assuming a prior over transitions model (even if such a prior is not updated afterwards in their approach). A multi-armed bandit approach is used to identify efficient policies in a space of formula-based policies, each policy being associated with an arm.

The optimistic principle has also already been proposed in the context of BRL. For instance, the BEB algorithm (for "Bayesian Exploration Bonus", see [26]) is a model-based approach that chooses actions according to the current expected model plus an additional reward bonus for state-action pairs that have been observed relatively little. The idea of adding such an exploration bonus is also proposed in the BVR algorithm (for "Bounded Variance Reward", see [35]) using a different type of bonuses. The BOSS algorithm (for "Best Of Sampled Set", see [2]) proposes a Thompson-like approach by (i) sampling models from a posterior distribution over transition models and (ii) combining the models into an optimistic MDP for decision making. A more efficient variant using an adaptive sampling process of the BOSS algorithm was also proposed in [11]. More recently, the BOLT algorithm (for "Bayesian Optimistic Local Transitions", see [1]) also adopts an optimistic principle by following a policy which is optimal with respect to an optimistic variant of the current expected model (obtained by adding artificial optimistic transitions). Even more recently, the BAMCP algorithm (for "Bayes-Adaptive Monte Carlo Planning", see [21]) proposes a UCT-like sparse sampling methods for Bayes-adaptive planning wich manages to achieve empirical state-of-the-art performance.

Like all methods listed in the previous paragraph, the BOP algorithm stands within the class of methods that make use of *optimism in the face of uncertainty* in the context of model-based BRL. Unlike these methods, the BOP algorithm proposes a tractable belief-lookahead approach in the sense that the belief is updated during the planning phase. This ensures that, whatever the number of transitions observed so-far, BOP converges towards Bayesian optimality as the budget parameter converges towards infinity.

III. Problem formalization

We first formalize the standard Reinforcement Learning (RL) problem in Section III-A. In Section III-B, we focus on the model-based Bayesian RL problem that we instantiate using Dirichlet distributions in Section III-C.

A. Reinforcement Learning

Let $M = (\mathcal{S}, \mathcal{A}, T, R)$ be a Markov Decision Process (MDP), where the set $\mathcal{S} = \left\{s^{(1)}, \dots, s^{(n_\mathcal{S})}\right\}$ denotes the finite state space and the set $\mathcal{A} = \left\{a^{(1)}, \dots, a^{(n_\mathcal{A})}\right\}$ the finite action space of the MDP. When the MDP is in state $s_t \in \mathcal{S}$ at time $t \in \mathbb{N}$, an action $a_t \in \mathcal{A}$ is selected and the MDP moves toward a next state $s_{t+1} \in \mathcal{S}$ drawn according to a probability

$$T(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t) \ .$$

It also receives a instantaneous deterministic scalar reward $r_t \in [0, 1]$:

$$r_t = R(s_t, a_t, s_{t+1}) \ .$$

In this paper, we assume that the transition model T is unknown. For simplicity, we assume that the value $R(s, a, s') \in [0, 1]$ is known for any possible transitions $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, which is often true in practice, e.g. in control R is often known to the user. Let $\pi : \mathcal{S} \to \mathcal{A}$ be a deterministic policy, i.e. a mapping from states to actions. A standard criterion for evaluating the performance of π is to consider its expected discounted return J^π defined as follows:

$$\forall s \in \mathcal{S}, \quad J^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1}) \big| s_0 = s\right]$$

where $\gamma \in [0, 1)$ is the so-called discount factor. An optimal policy is a policy π^* such that, for any policy π,

$$\forall s \in \mathcal{S}, \quad J^{\pi^*}(s) \geq J^{\pi}(s) \ .$$

Such an optimal policy π^* is scored with an optimal return $J^*(s) = J^{\pi^*}(s)$ which satisfies the Bellman optimality equation:

$$\begin{aligned}&\forall s \in \mathcal{S}, \\ &J^*(s) = \max_{a \in \mathcal{A}} \quad \sum_{s' \in \mathcal{S}} T(s, a, s') \left(R(s, a, s') + \gamma J^*(s') \right) \ .\end{aligned}$$

Finding an optimal policy can thus be theoretically achieved by behaving greedily with respect to the optimal state-action value function $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ defined as follows:

$$\begin{aligned}&\forall (s, a) \in \mathcal{S} \times \mathcal{A}, \\ &Q^*(s, a) = \sum_{s' \in \mathcal{S}} T(s, a, s') \left[R(s, a, s') + \gamma J^*(s') \right] \ .\end{aligned}$$

One major difficulty in our setting resides in the fact that the transition model $T(\cdot, \cdot, \cdot)$ is initially unknown and need to be learned through interactions. This implicitly leads to a trade-off between acting optimally with respect to the current knowledge of the unknown transition model (exploitation) and acting in order to increase the knowledge about the unknown transition model (exploration).

B. Model-based Bayesian Reinforcement Learning

Model-based Bayesian RL proposes to address the exploration/exploitation (E/E) trade-off by representing the knowledge about the unknown transition model using a probability distribution over all possible transition models $\boldsymbol{\mu}$. In this setting, an initial prior distribution $\boldsymbol{b}_0$ is given and iteratively updated according to the Bayes rule as new samples of the actual transition model are generated. At any time-step t, the so-called posterior distribution $\boldsymbol{b}_t$ depends on the prior distribution $\boldsymbol{b}_0$ and the history $h_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$ observed so-far. The Markovian property implies that the posterior $\boldsymbol{b}_{t+1}$:

$$\boldsymbol{b}_{t+1} = P(\boldsymbol{\mu} | h_{t+1}, \boldsymbol{b}_0)$$

can be updated sequentially:

$$\boldsymbol{b}_{t+1} = P(\boldsymbol{\mu} | (s_t, a_t, s_{t+1}), \boldsymbol{b}_t) \ .$$

The posterior distribution $\boldsymbol{b}_t$ over all possible models is called "belief" in the Bayesian RL literature.

A standard approach to $-$ theoretically $-$ solve Bayesian RL problems is to consider a BA-state $\boldsymbol{z}$ obtained by concatenating the state with the belief $\boldsymbol{z} = \langle s, \boldsymbol{b} \rangle$ and solving the corresponding BA-MDP [17], [15]. In the following, we denote by $\mathbb{B}$ the BA-state space. This BA-MDP is defined by a transition function $\mathbf{T}$ given by:

$$\begin{aligned}&\forall (\boldsymbol{z}, \boldsymbol{z}') \in \mathbb{B}^2, \forall a \in \mathcal{A}, \\ &\qquad \mathbf{T}(\boldsymbol{z}, a, \boldsymbol{z}') &=& \ P(\boldsymbol{z}' | (\boldsymbol{z}, a)) \\ && =& \ P(\boldsymbol{b}' | \boldsymbol{b}, s, a, s') \mathbb{E}\left[P(s' | s, a) | \boldsymbol{b} \right] \\ && =& \ \mathbf{1}_{\{h_{t+1} = (h_t, a, s')\}} \mathbb{E}\left[P(s' | s, a) | \boldsymbol{b} \right]\end{aligned}$$

and a reward function $\mathbf{R}$ given by:

$$\forall (\boldsymbol{z}, \boldsymbol{z}') \in \mathbb{B}^2, \forall a \in \mathcal{A}, \quad \mathbf{R}(\boldsymbol{z}, a, \boldsymbol{z}') = R(s, a, s') \ .$$

A Bayesian optimal policy $\boldsymbol{\pi}^*$ can be theoretically obtained by behaving greedily with respect to the optimal Bayesian state-action value function $\mathbf{Q}^*$:

$$\forall \boldsymbol{z} \in \mathbb{B}, \quad \boldsymbol{\pi}^*(\boldsymbol{z}) = \underset{a \in \mathcal{A}}{\arg\max} \quad \mathbf{Q}^*(\boldsymbol{z}, a)$$

where $\forall \boldsymbol{z} \in \mathbb{B}, \forall a \in \mathcal{A}$,

$$\mathbf{Q}^*(\boldsymbol{z}, a) = \sum_{\boldsymbol{z}'} \mathbf{T}\left(\boldsymbol{z}, a, \boldsymbol{z}' \right) \left(\mathbf{R}(\boldsymbol{z}, a, \boldsymbol{z}') + \gamma \mathbf{J}^*(\boldsymbol{z}') \right) \ .$$

Here, $\boldsymbol{z}'$ are reachable belief state when taking action a in belief state $\boldsymbol{z}$ and $\mathbf{J}^*(\boldsymbol{z})$ is the Bayesian optimal return:

$$\mathbf{J}^*(\boldsymbol{z}) = \max_{a \in \mathcal{A}} \quad \mathbf{Q}^*(\boldsymbol{z}, a) \ .$$

In this work, the goal is to take decisions that are near-optimal in the Bayesian meaning, i.e. we want to find a policy which is as close as possible as $\boldsymbol{\pi}^*$.

C. Dirichet distribution-based BRL

One needs to define a class of distributions. A most usual approach is to consider one independent Dirichlet distribution for each state-action transition. We obtain a posterior $\boldsymbol{b}$ whose probability density function is:

$$d(\boldsymbol{\mu}; \boldsymbol{\Theta}) = \prod_{(s,a) \in \mathcal{S} \times \mathcal{A}} D\left(\boldsymbol{\mu}_{s,a}; \boldsymbol{\Theta}(s, a, \cdot) \right)$$

where $D(\cdot; \cdot)$ denotes a Dirichlet distribution, $\boldsymbol{\Theta}(s, a, s')$ denotes the number of observed transitions from $(s, a) \in \mathcal{S} \times \mathcal{A}$ towards every $s' \in \mathcal{S}$ and $\boldsymbol{\Theta}(s, a, \cdot)$ denotes the vector of counters of observed transitions:

$$\boldsymbol{\Theta}(s, a, .) = \left[\boldsymbol{\Theta}\left(s, a, s^{(1)} \right), \ldots, \boldsymbol{\Theta}\left(s, a, s^{(n_{\mathcal{S}})} \right) \right]$$

and $\boldsymbol{\Theta}$ is the matrix that contains all $\boldsymbol{\Theta}(s, a, .) \quad s \in \mathcal{S}, a \in \mathcal{A}$. In the following, we denote by $\boldsymbol{b}\left(\boldsymbol{\Theta} \right)$ such a Dirichlet distribution-based posterior. The resulting posterior distribution $\boldsymbol{b}\left(\boldsymbol{\Theta} \right)$ satisfies the following well-known property:

$$\mathbb{E}\left[P(s' | s, a) | \boldsymbol{b}\left(\boldsymbol{\Theta} \right) \right] = \frac{\boldsymbol{\Theta}(s, a, s')}{\sum_{s'' \in \mathcal{S}} \boldsymbol{\Theta}(s, a, s'')}$$

and the Bayesian update under the observation of a transition $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is reduced to a simple increment of the corresponding counter:

$$\boldsymbol{\Theta}(s, a, s') \leftarrow \boldsymbol{\Theta}(s, a, s') + 1 \ .$$

In such a context, the Bayesian optimal state-action value function writes:

$$\begin{aligned}\mathbf{Q}^*(\langle s, \boldsymbol{b}\left(\boldsymbol{\Theta} \right) \rangle, a) \quad = \quad & \sum_{s' \in \mathcal{S}} \frac{\boldsymbol{\Theta}(s, a, s')}{\sum_{s'' \in \mathcal{S}} \boldsymbol{\Theta}(s, a, s'')} \Big(R(s, a, s') \\ & + \gamma \mathbf{J}^*(\langle s', \boldsymbol{b}\left(\boldsymbol{\Theta}'_{s,a,s'} \right) \rangle) \Big)\end{aligned}$$

where $\boldsymbol{\Theta}'_{s,a,s'}$ is such that:

$$\boldsymbol{\Theta}'_{s,a,s'}(x, y, x') = \begin{cases} \boldsymbol{\Theta}(x, y, x') + 1 \text{ if } (x, y, x') = (s, a, s'), \\ \qquad \boldsymbol{\Theta}(x, y, x') \text{ otherwise.} \end{cases}$$

IV. The BOP algorithm

In this section, we describe our contribution, the *Bayesian Optimistic Planning* (BOP) algorithm. We first formalize the notion of *BA-planning trees* in Section IV-A. The BOP algorithm is based on an optimistic approach for expanding a BA-planning tree that we detail in Section IV-B.

A. BA-planning trees

Each node in a BA-planning tree is denoted by $\boldsymbol{x}$ and labeled by a BA-state $\boldsymbol{z} = \langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$. Many nodes may have the same label $\boldsymbol{z}$, for this reason we distinguish nodes from their belief states labels. A node $\boldsymbol{x}$ is extended by adding to it, for each action $a \in \mathcal{A}$, and then for each $\boldsymbol{z}' = \left\langle s', \boldsymbol{b}\left(\boldsymbol{\Theta}'_{s,a,s'}\right)\right\rangle$, a child node $\boldsymbol{x}'$ labeled by $\boldsymbol{z}'$. The branching factor of the tree is thus $n_{\mathcal{S}} \times n_{\mathcal{A}}$. Let us denote by $\mathcal{C}(\boldsymbol{x}, a)$ the set of children $\boldsymbol{x}'$ corresponding to action a, and by $\mathcal{C}(\boldsymbol{x})$ the union:

$$\mathcal{C}(\boldsymbol{x}) = \bigcup_{a \in \mathcal{A}} \mathcal{C}(\boldsymbol{x}, a) \ .$$

B. Optimistic planning in a BA-state space

The BOP algorithm builds a belief-augmented planning tree starting from a root node that contains the belief state where an action has to be chosen. At each iteration, the algorithm actively selects a leaf of the tree and expands it by generating, for every action, all possible successor belief-augmented states. The algorithm stops growing the tree after a fixed *expansion budget* $n \in \mathbb{N} \setminus \{0\}$ and returns an action on the basis of the final tree. The heart of this approach is the procedure to select leaves for expansion. To this end, we design an optimistic strategy that assumes the best possible optimal values compatible with the belief-augmented planning tree generated so far. To formalize this optimistic strategy, let us first introduce some notations;

- The entire tree is denoted by $\mathcal{T}$, and the set of leaf nodes by $\mathcal{L}(\mathcal{T})$;
- A node of the tree $\boldsymbol{x}$ is labeled with its associated belief-augmented state $\boldsymbol{z} = \langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$. A child node is denoted by $\boldsymbol{x}'$ (and labeled by $\boldsymbol{z}' = \left\langle s', \boldsymbol{b}\left(\boldsymbol{\Theta}'_{s,a,s'}\right)\right\rangle$ where a is the action that was taken to jump from $\boldsymbol{z}$ to $\boldsymbol{z}'$) and is also called next state.
- The depth of a node $\boldsymbol{x}$ is denoted by $\Delta(\boldsymbol{x})$.

An illustration of a belief-augmented planning tree is given in Figure 1.

Expansion criterion. For each $\boldsymbol{x} \in \mathcal{T}$ (labeled by $\boldsymbol{z} = \langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$) and $a \in \mathcal{A}$, we recursively define the B-values $B(\boldsymbol{x}, a)$ as follows:

$$\forall \boldsymbol{x} \in \mathcal{L}(\mathcal{T}), \forall a \in \mathcal{A} \ , B(\boldsymbol{x}, a) = \frac{1}{1-\gamma},$$
$$\forall \boldsymbol{x} \in \mathcal{T} \setminus \mathcal{L}(\mathcal{T}), \forall a \in \mathcal{A} \ , B(\boldsymbol{x}, a) =$$
$$\sum_{\boldsymbol{x}' \in \mathcal{C}(\boldsymbol{x}, a)} \mathbf{T}(\boldsymbol{z}, a, \boldsymbol{z}') \left(\mathbf{R}(\boldsymbol{z}, a, \boldsymbol{z}') + \gamma \max_{a' \in \mathcal{A}} \quad B(\boldsymbol{x}', a') \right) .$$

Each B-value $B(\boldsymbol{x}, a)$ is an upper bound for the optimal Bayesian state-action value function $\mathbf{Q}^*(\langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle, a)$.

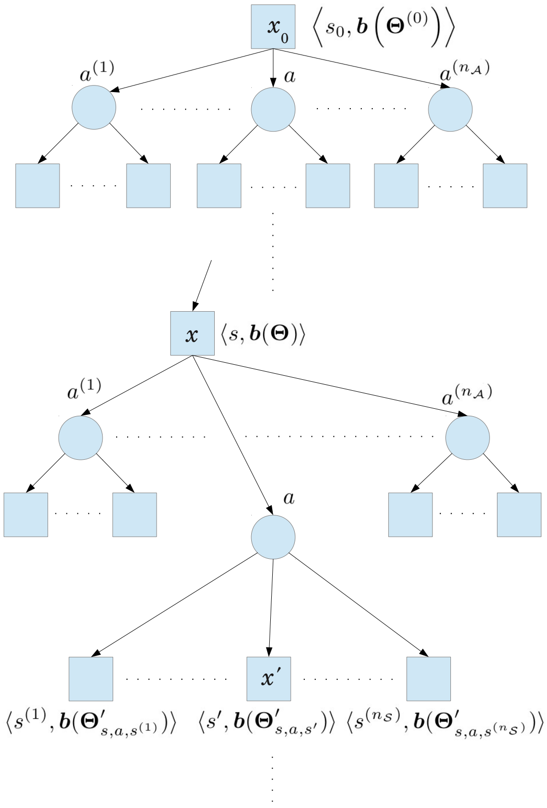
x_0
$\left\langle s_0, \boldsymbol{b}\left(\boldsymbol{\Theta}^{(0)}\right)\right\rangle$
$a^{(1)}$
a
$a^{(n_{\mathcal{A}})}$
x
$\langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$
$a^{(1)}$
$a^{(n_{\mathcal{A}})}$
a
x'
$\langle s^{(1)}, \boldsymbol{b}(\boldsymbol{\Theta}'_{s,a,s^{(1)}})\rangle$
$\langle s', \boldsymbol{b}(\boldsymbol{\Theta}'_{s,a,s'})\rangle$
$\langle s^{(n_S)}, \boldsymbol{b}(\boldsymbol{\Theta}'_{s,a,s^{(n_S)}})\rangle$

Fig. 1. Illustration of a BA-planning tree. Squares are BA-state nodes whereas circles represent decisions.

To obtain a set of candidate leaf nodes for expansion, we build an optimistic subtree by starting from the root and selecting at each node only its children that are associated to optimistic actions:

$$a^{\dagger}(\boldsymbol{x}) \in \arg\max_{a \in \mathcal{A}} \quad B(\boldsymbol{x}, a)$$

(here ties are broken always the same). We denote by $\mathcal{T}^{\dagger}$ and $\mathcal{L}\left(\mathcal{T}^{\dagger}\right)$ the resulting optimistic subtree and its corresponding set of leaves. An illustration of such an optimistic subtree is given in Figure 2

To choose one leaf node to expand among the candidates $\mathcal{L}\left(\mathcal{T}^{\dagger}\right)$, we propose to maximize the potential decrease of the B-value at the root of the belief state tree $B\left(\boldsymbol{x}_0, a^{\dagger}(\boldsymbol{x}_0)\right)$. Such a B-value can be written more explicitly as an expected optimistic return obtained along the paths from the root to all the leaf nodes in the optimistic subtree:

$$B(\boldsymbol{x}_0, a^{\dagger}(\boldsymbol{x}_0)) = \sum_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}^{\dagger})} \mathbf{P}(\boldsymbol{x}) \left(\bar{\mathbf{R}}(\boldsymbol{x}) + \frac{\gamma^{\Delta(\boldsymbol{x})}}{1-\gamma} \right)$$

where $\mathbf{P}(\boldsymbol{x})$ is the probability to reach $\boldsymbol{x} \in \mathcal{L}\left(\mathcal{T}^{\dagger}\right)$ (product of probabilities along the path) and $\bar{\mathbf{R}}(\boldsymbol{x})$ is the discounted sum or reward gathered along the path. If we denote the path by $\boldsymbol{y}_0^{\boldsymbol{x}}, \boldsymbol{y}_1^{\boldsymbol{x}}, \ldots, \boldsymbol{y}_{\Delta(\boldsymbol{x})}^{\boldsymbol{x}}$ for a given $\boldsymbol{x}$ and $\boldsymbol{z}_0^{\boldsymbol{x}}, \boldsymbol{z}_1^{\boldsymbol{x}}, \ldots, \boldsymbol{z}_{\Delta(\boldsymbol{x})}^{\boldsymbol{x}}$ the associated sequence of labels ($\boldsymbol{y}_0^{\boldsymbol{x}} = \boldsymbol{x}_0$ and $\boldsymbol{y}_{\Delta(\boldsymbol{x})}^{\boldsymbol{x}} = \boldsymbol{x}$), we obtain:

$$\begin{aligned}\mathbf{P}(\boldsymbol{x}) &= \prod_{d=0}^{\Delta(\boldsymbol{x})-1} \mathbf{T}\left(\boldsymbol{z}_d^{\boldsymbol{x}}, a^\dagger\left(\boldsymbol{y}_d^{\boldsymbol{x}}\right), \boldsymbol{z}_{d+1}^{\boldsymbol{x}}\right)\\ \bar{\mathbf{R}}(\boldsymbol{x}) &= \sum_{d=0}^{\Delta(\boldsymbol{x})-1} \gamma^d \mathbf{R}\left(\boldsymbol{z}_d^{\boldsymbol{x}}, a^\dagger\left(\boldsymbol{y}_d^{\boldsymbol{x}}\right), \boldsymbol{z}_{d+1}^{\boldsymbol{x}}\right)\end{aligned}$$

($\mathbf{P}$ and $\bar{\mathbf{R}}$ are both defined on nodes). Consider the contribution of a single leaf node to Equation 1:

$$\mathbf{P}(\boldsymbol{x})\left(\bar{\mathbf{R}}(\boldsymbol{x}) + \frac{\gamma^{\Delta(\boldsymbol{x})}}{1-\gamma}\right) .$$

If this leaf node were expanded, its contribution would decrease the most if the rewards along the transitions to all the new children were 0. In that case, its updated contribution would be $\mathbf{P}(\boldsymbol{x})\left(\bar{\mathbf{R}}(\boldsymbol{x}) + \frac{\gamma^{\Delta(\boldsymbol{x})+1}}{1-\gamma}\right)$, and its contribution would have decreased by:

$$\mathbf{P}(\boldsymbol{x})\left(\bar{\mathbf{R}}(\boldsymbol{x}) + \frac{\gamma^{\Delta(\boldsymbol{x})}}{1-\gamma} - \bar{\mathbf{R}}(\boldsymbol{x}) - \frac{\gamma^{\Delta(\boldsymbol{x})+1}}{1-\gamma}\right) = \mathbf{P}(\boldsymbol{x})\gamma^{\Delta(\boldsymbol{x})} .$$

So, finally, the rule for selecting a node to expand $\boldsymbol{x}_e$ is the following:

$$\boldsymbol{x}_e \in \underset{x \in \mathcal{L}(\mathcal{T}^\dagger)}{\arg\max} \quad \mathbf{P}(\boldsymbol{x})\ \gamma^{\Delta(\boldsymbol{x})} .$$

Algorithm 1 The BOP algorithm.

```
input initial belief state $\boldsymbol{z}_0 = \left\langle s_0, \boldsymbol{b}\left(\boldsymbol{\Theta}^{(0)}\right)\right\rangle$;
        a budget parameter $n$;
output a near-Bayesian optimal action $\tilde{a}_n(\boldsymbol{z}_0)$;
initialize $\mathcal{T}_0 \leftarrow \{\boldsymbol{x}_0\}$;
for $t = 0, \ldots, n-1$ do
    starting from $\boldsymbol{x}_0$, build the optimistic subtree $\mathcal{T}_t^\dagger$;
    select leaf to expand: $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}_t^\dagger)} \ \mathbf{P}(\boldsymbol{z})\gamma^{\Delta(\boldsymbol{x})}$;
    expand $\boldsymbol{x}_t$ and obtain $\mathcal{T}_{t+1}$;
end for
return $\tilde{a}_n(\boldsymbol{z}_0) \in \arg\max_{a \in \mathcal{A}} \ \nu(\boldsymbol{x}_0, a)$
run action $\tilde{a}_n(\boldsymbol{z}_0)$;
    observe a subsequent state $\tilde{s}$;
update the initial vector of counters:
    $\boldsymbol{\Theta}^{(0)}(s_0, \tilde{a}_n(\boldsymbol{z}_0), \tilde{s}) \leftarrow \boldsymbol{\Theta}^{(0)}(s_0, \tilde{a}_n(\boldsymbol{z}_0), \tilde{s}) + 1$
```

Action selection at the root. Similarly to the B-values, we define the ν-values:

$$\begin{aligned}&\forall \boldsymbol{x} \in \mathcal{L}(\mathcal{T}), \forall a \in \mathcal{A}\ , \nu(\boldsymbol{x}, a) = 0,\\ &\forall \boldsymbol{x} \in \mathcal{T} \setminus \mathcal{L}(\mathcal{T}), \forall a \in \mathcal{A}\ , \nu(\boldsymbol{x}, a) =\\ &\sum_{\boldsymbol{x}' \in \mathcal{C}(\boldsymbol{x}, a)} \mathbf{T}\left(\boldsymbol{z}, a, \boldsymbol{z}'\right)\left(\mathbf{R}\left(\boldsymbol{z}, a, \boldsymbol{z}'\right) + \gamma \max_{a' \in \mathcal{A}} \quad \nu(\boldsymbol{x}', a')\right)\end{aligned}$$

The difference from the B-values is that its starts with 0 values at the leaves. At the end, the root action $\tilde{a}_n(\boldsymbol{z}_0)$ is selected as follows:

$$\tilde{a}_n(\boldsymbol{z}_0) \in \underset{a' \in \mathcal{A}}{\arg\max} \quad \nu(\boldsymbol{x}_0, a') .$$

Maximizing the lower bound $\nu(\boldsymbol{x}_0, \cdot)$ can be seen as taking a cautious decision. We give in Table 1 a tabular version of the BOP algorithm.

Finally, observe that the branching factor of the belief-augmented planning trees is $n_{\mathcal{S}} \times n_{\mathcal{A}}$, which is equal to the branching factor of the planning trees used in the original OP-MDP algorithm. The only additional complexity of the BOP algorithm is that one needs to propagate and update the counter $\boldsymbol{\Theta}$ in the belief-augmented planning tree. Also note that in real-life applications, it is often the case that the set of reachable states starting from a given state is much smaller than $\mathcal{S}$. If such a priori knowledge is available, it can be exploited by BOP, and the branching factor becomes $n'_{\mathcal{S}} \times n_{\mathcal{A}}$ with $n'_{\mathcal{S}} \ll n_{\mathcal{S}}$.

V. Reinterpretation of the BOP algorithm

In this section, we reinterprete the BOP algorithm similarly to [9] as a branch-and-bound-type optimization in the space of *BA-planning tree-policies* (tree-policies for short). A tree-policy h is an assignment of actions to a subtree $\mathcal{T}_h$ of the infinite belief-augmented planning tree $\mathcal{T}_\infty$:

$$h : \mathcal{T}_h \rightarrow \mathcal{A},$$

recursively taking into account only the nodes reached under the action choices made so far:

$$\mathcal{T}_h = \{\boldsymbol{x} \in \mathcal{T}_\infty | \boldsymbol{x} = \boldsymbol{x}_0 \text{ or } \exists \boldsymbol{x}' \in \mathcal{T}_h, \boldsymbol{x} \in \mathcal{C}(\boldsymbol{x}', h(\boldsymbol{x}'))\ \}$$

where actions $h(\boldsymbol{x})$ are assigned as desired. The branching factor of $\mathcal{T}_h$ is at most $n_{\mathcal{S}}$. Denote the Bayesian expected return of the tree-policy h by $\mathbf{v}(h)$, and the optimal, maximal Bayesian return by $\mathbf{v}^*$.

A class of tree-policies, $H : \mathcal{T}_H \rightarrow \mathcal{A}$, is obtained similarly but restricting the procedure to nodes in some finite tree $\mathcal{T}_t$ considered by BOP, so that all action assignments below $\mathcal{T}_t$ are free. H is a set of tree-policies, where one such tree-policy $h \in H$ is obtained by initializing the free actions. Note that $\mathcal{T}_H = \mathcal{T}_t \cap \mathcal{T}_h$ for any $h \in H$. Note that tree-policies h are more general than the usual stationary, deterministic policies that would always take the same action in a given belief-augmented state $\boldsymbol{z}$.

The expected Bayesian return of any tree-policy h belonging to some class H is lower-bounded by:

$$\nu_H = \sum_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}_H)} \mathbf{P}(\boldsymbol{x})\bar{\mathbf{R}}(\boldsymbol{x})$$

because the rewards that h can obtain below the leaves of $\mathcal{L}(\mathcal{T}_H)$ are lower-bounded by 0. Since rewards are also upper-

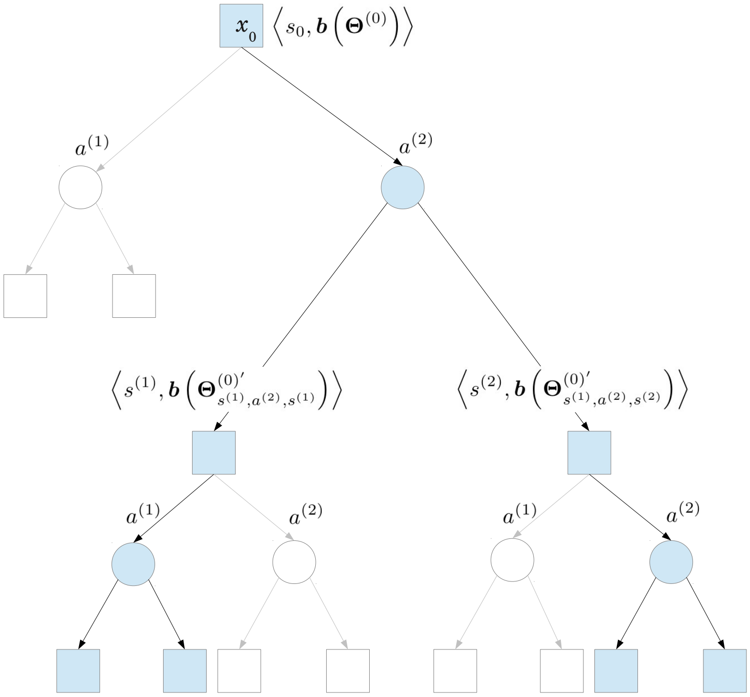
x_0
$\left\langle s_0, \boldsymbol{b}\left(\boldsymbol{\Theta}^{(0)}\right)\right\rangle$
$a^{(1)}$
$a^{(2)}$
$\left\langle s^{(1)}, \boldsymbol{b}\left(\boldsymbol{\Theta}^{(0)'}_{s^{(1)},a^{(2)},s^{(1)}}\right)\right\rangle$
$\left\langle s^{(2)}, \boldsymbol{b}\left(\boldsymbol{\Theta}^{(0)'}_{s^{(1)},a^{(2)},s^{(2)}}\right)\right\rangle$
$a^{(1)}$
$a^{(2)}$
$a^{(1)}$
$a^{(2)}$

Fig. 2. Illustration of an optimistic subtree in the case $n_{\mathcal{S}} = n_{\mathcal{A}} = 2$. Parts of the original tree that do not belong to the optimistic subtree are in light gray / white.

bounded by 1, an upper bound on the value of $h \in H$ is:

$$\begin{aligned} B_H &= \sum_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}_H)} \mathbf{P}(\boldsymbol{x}) \left[\bar{\mathbf{R}}(\boldsymbol{x}) + \frac{\gamma^{\Delta(\boldsymbol{x})}}{1-\gamma} \right] \\ &= \nu_H + \sum_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}_H)} c(\boldsymbol{x}) = \nu_H + \operatorname{diam}(H) \end{aligned}$$

where we introduce the notations:

$$c(\boldsymbol{x}) = \mathbf{P}(\boldsymbol{x}) \frac{\gamma^{\Delta(\boldsymbol{x})}}{1-\gamma},$$

the contribution of a leaf $\boldsymbol{x}$ to the difference between the upper and lower bounds, and

$$\operatorname{diam}(H) = \sum_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}_H)} c(\boldsymbol{x})$$

the diameter of H. Note that $\operatorname{diam}(H) = \sup_{h,h' \in H} \delta(h, h')$ where δ is a metric defined over the space of tree-policies:

$$\delta(h, h') = \sum_{\boldsymbol{x} \in \mathcal{L}(\mathcal{T}_h \cap \mathcal{T}_{h'})} c(\boldsymbol{x}) \ .$$

Using these notations, BOP can be reformulated as follows. At each iteration, the algorithm selects an optimistic tree-policy class which maximizes the upper bound among all classes compatible with the current tree $\mathcal{T}_t$:

$$H_t^\dagger \in \operatorname*{arg\,max}_{H \in \mathcal{T}_t} \quad B_H$$

where $H \in \mathcal{T}_t$ means that $\mathcal{T}_H \subseteq \mathcal{T}_t$. The optimistic class is explored deeper, by expanding one of its leaf nodes (making the action choices for that note definite). The chosen leaf is the one maximizing the contributions $c(\boldsymbol{x})$ to the uncertainty $\operatorname{diam}\left(H_t^\dagger\right)$ on the value of policies $h \in H_t^\dagger$:

$$\boldsymbol{x}_t \in \operatorname*{arg\,max}_{\boldsymbol{x} \in \mathcal{L}\left(\mathcal{T}_{H_t^\dagger}\right)} \quad c(\boldsymbol{x}) \ .$$

Under the metric δ, this can also be seen as splitting the set of tree policies H along the longest edge, where H is a hyperbox with $\left|\mathcal{L}\left(\mathcal{T}_{H_t^\dagger}\right)\right|$ dimensions, having a length of $c(\boldsymbol{x})$ along dimension $\boldsymbol{x}$. The algorithm continues at the next iteration with the new, resulting tree $\mathcal{T}_{t+1}$. After n iterations, a policy class is chosen, by maximizing the lower bound :

$$H_n^* \in \operatorname*{arg\,max}_{H \in \mathcal{T}_n} \quad \nu_H \ .$$

The action $\tilde{a}_n(\boldsymbol{z}_0)$ returned by BOP is then the first action chosen by H_n^*.

VI. Theoretical results

Let $\mathcal{R}_n(\boldsymbol{z}_0)$ be the Bayesian simple regret:

$$\mathcal{R}_n(\boldsymbol{z}_0) = \mathbf{J}^*(\boldsymbol{z}_0) - \mathbf{Q}^*(\boldsymbol{z}_0, \tilde{a}_n(\boldsymbol{z}_0)) \ ,$$

i.e. the loss - with respect to the Bayesian optimal policy - of taking action $\tilde{a}_n(\boldsymbol{z}_0))$ instead of $\boldsymbol{\pi}^*(\boldsymbol{z}_0)$. We have the result:

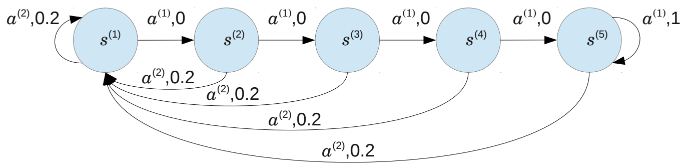

Fig. 3. The standard 5-state chain problem.

Algorithm	Performance
BEB ($\beta = 150$) [26]	165.2
BEETLE [32]	175.4
BOP ($n = 50$)	**255.6**
BOLT ($\eta = 150$) [1]	278.7
BOLT ($\eta = 7$) [1]	289.6
BOP ($n = 100$)	**292.9**
BOSS [2]	300.3
BOP ($n = 200$)	**304.6**
EXPLOIT [32]	307.8
BOP ($n = 500$)	**308.8**
BEB ($\beta = 1$) [26]	343.0
BVR [35]	346.5
Optimal strategy	**367.7**

TABLE I
PERFORMANCE OF BOP COMPARED WITH OTHER MODEL-BASED BRL APPROACHES ON THE FULL-PRIOR 5-STATE CHAIN MDP PROBLEM.

Theorem: For any BA-state $\boldsymbol{z}_0 \in \mathbb{B}$, there exists a *near-optimality exponent* $\beta(\boldsymbol{z}_0) \in \mathbb{R}^+$ such that

$$\mathcal{R}_n(\boldsymbol{z}_0) = \tilde{O}\left(n^{-\frac{1}{\beta(\boldsymbol{z}_0)}}\right) \text{ if } \beta(\boldsymbol{z}_0) > 0,$$

and when $\beta(\boldsymbol{z}_0) = 0$, the regret is exponentially decreasing with n. It follows that:

$$\forall \boldsymbol{z}_0 \in \mathbb{B}, \quad \lim_{n \to \infty} \quad \mathcal{R}_n(\boldsymbol{z}_0) = 0 \ .$$

This result directly follows from the analysis of the OP-MDP algorithm [9], that we apply here in the context of a BA-MDP (which is also a MDP). The near-optimality exponent $\beta(\boldsymbol{z}_0)$ measures the rate of growth of a certain set of important nodes of the BA-planning tree rooted at $\boldsymbol{z}_0$: roughly speaking, nodes that make large contributions to near-Bayes optimal policies. $\beta(\cdot)$ varies from 0, corresponding to the slowest growth (easy planning problem), to $\ln(n_{\mathcal{A}} n_{\mathcal{S}}) / \ln(1/\gamma)$, corresponding to the fastest growth (difficult planning problem).

As the number of observed transitions goes to infinity, the distribution over transition models converges towards a Dirac centered on the actual MDP, and we conjecture that the $\beta(\boldsymbol{z}_0)$ should converge towards the parameter $\beta(s_0)$ of the underlying MDP, meaning that the complexity of planning in the BA-MDP becomes similar to the complexity of planning in the underlying MDP.

VII. Experimental illustration

We compare the BOP algorithm with other model-based Bayesian RL algorithms on the standard 5-state chain problem [36] which is one of the most usual benchmarks for evaluating BRL algorithms. In this benchmark, the state space contains 5 states ($n_{\mathcal{S}} = 5$), and two actions are possible ($n_{\mathcal{A}} = 2$). Taking action $a^{(1)}$ in state $s^{(i)}$ leads to jump towards state $s^{(i+1)}$, except in state $s^{(5)}$ where it makes the agent stay in $s^{(5)}$ and receive a +1 reward. Taking action $a^{(2)}$ makes the agent go back to state 1 and get a reward of .2. With probability $p = .2$, taking an action has the effect of the other action. The optimal strategy is to take action 1 whatever the state. An illustration is given in Figure 3.

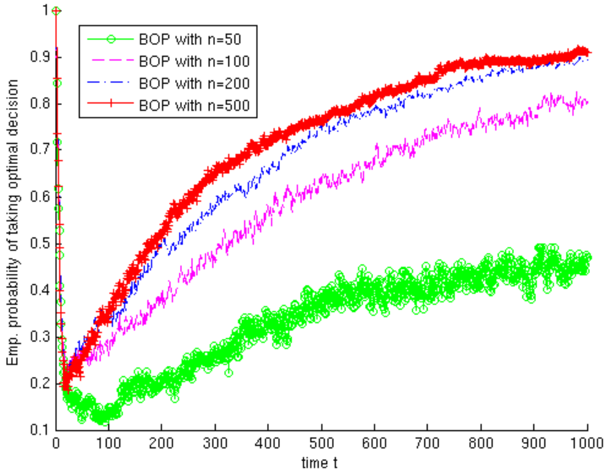

Fig. 4. Empirical probability of taking the optimal decision (action $a^{(1)}$) over time (note that action $a^{(1)}$ is optimal for all states).

The transition model is unknown to the agent. In our experiments, we consider a full prior which means that we do not incorporate any specific prior knowledge (all transitions are possible). In the particular context of Dirichlet distributions, the full prior hypothesis is implemented by initializing $\boldsymbol{\Theta}^{(0)}$ as follows: $\forall (s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \quad \boldsymbol{\Theta}^{(0)}(s, a, s') = 1$.

We have run 500 times the BOP algorithm starting from state $s_0 = 1$ and applying BOP decisions during 1000 time-steps for four different values of the budget parameter $n \in \{50, 100, 200, 500\}$. The empirical average performance (in terms of cumulative undiscounted received rewards) of the BOP algorithm are give in Table I. Standard error is on the order of 2 to 5. We also display in Table I the performances obtained by other BRL algorithms in the very same settings (obtained from the literature).

We first observe that the performances of the BOP algorithm increase with n. Then, we observe that the BOP algorithm with $n = 500$ offers performances that are better than other algorithms, except those using exploration bonuses such as BEB (with a tuned value of its parameter β) and BVR which outperform the BOP algorithm on this benchmark. Do not forget that Bayesian optimality differs from optimality in the underlying MDP, so it is not suprising that some algorithms may be here more efficient than BOP, which is nevertheless likely to be close to Bayesian optimality with $n = 500$. We also display in Figure 4 the evolution over time of the empirical probability (computed over the 500 runs) that the BOP algorithm takes optimal decision for $n \in \{50, 100, 200, 500\}$. For information, the computation of one 1000 time-steps run of the BOP algorithm takes about 10 hours (resp. 1 hour, 20 minutes and 5 minutes) on a standard recent one-core linux machine with $n = 500$ (resp. $n = 200$, $n = 100$ and $n = 50$) using Matlab®.

VIII. Conclusions and future work

We have proposed BOP (for "Bayesian Optimistic Planning"), a new model-based Bayesian reinforcement learning algorithm that extends the principle of the OP-MDP algorithm [10], [9] to the context where the transition model is initially unknown and has to be learned.

In this paper, we have considered a finite state space, but one could extend BOP to infinite state space settings by constraining the branching factor of the belief-augmented planning tree. Another open and interesting research direction is to analyze how the near-optimality exponent of the belief-augmented MDP relates to the exponent of the underlying MDP.

Acknowledgments

Raphael Fonteneau is a post-doctoral fellow of the F.R.S - FNRS. Lucian Buşoniu is a research scientist with CNRS. We also thank the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreements n°216886 (PASCAL2) and n°270327 (CompLACS) and the Belgian Network DYSCO funded by the IAP Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s). We also thank Olivier Nicol for his precious help.

References

[1] M. Araya, V. Thomas, and O. Buffet. Near-optimal BRL using optimistic local transitions. In *International Conference on Machine Learning (ICML)*, 2012.

[2] J. Asmuth, L. Li, M.L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *Uncertainty in Artificial Intelligence (UAI)*, pages 19–26, 2009.

[3] J. Asmuth and ML Littman. Approaching Bayes-optimalilty using Monte-Carlo tree search. In *International Conference on Automated Planning and Scheduling (ICAPS), Freiburg, Germany*, 2011.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite time analysis of multiarmed bandit problems. *Machine Learning*, 47:235–256, 2002.

[5] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[6] R.I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2003.

[7] S. Bubeck and R. Munos. Open loop optimistic planning. In *Conference on Learning Theory (COLT)*, pages 477–489, 2010.

[8] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. Online optimization in X-armed bandits. In *Neural Information Processing Systems (NIPS)*, pages 201–208, 2009.

[9] L. Busoniu and R. Munos. Optimistic planning for markov decision processes. In *International Conference on Artificial Intelligence and Satistics (AISTATS), JMLR W & CP 22*, pages 182–189, 2012.

[10] L. Busoniu, R. Munos, B. De Schutter, and R. Babuska. Optimistic planning for sparsely stochastic systems. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 48–55, 2011.

[11] P. Castro and D. Precup. Smarter sampling in model-based Bayesian reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, pages 200–214, 2010.

[12] M. Castronovo, F. Maes, R. Fonteneau, and D. Ernst. Learning exploration/exploitation strategies for single trajectory reinforcement learning. In *European Workshop on Reinforcement Learning (EWRL)*, 2012.

[13] R. Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. *Computers and Games*, pages 72–83, 2007.

[14] R. Dearden, N. Friedman, and S. Russell. Bayesian Q-learning. In *National Conference on Artificial Intelligence*, pages 761–768, 1998.

[15] C. Dimitrakakis. Tree exploration for Bayesian RL exploration. In *International Conference on Computational Intelligence for Modelling Control & Automation*, pages 1029–1034, 2008.

[16] C. Dimitrakakis and M. G. Lagoudakis. Rollout sampling approximate policy iteration. *Machine Learning*, 72:157–171, 2008.

[17] M.O.G. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts Amherst, 2002.

[18] A.A. Feldbaum. Dual control theory. *Automation and Remote Control*, 21(9):874–1039, 1960.

[19] S. Gelly, Y. Wang, R. Munos, and O. Teytaud. Modification of UCT with patterns in Monte-Carlo go. Technical report, INRIA RR-6062, 2006.

[20] J.C. Gittins. *Multiarmed Bandit Allocation Indices*. Wiley, 1989.

[21] A. Guez, D. Silver, and P. Dayan. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *Neural Information Processing Systems (NIPS)*, 2012.

[22] J.F. Hren and R. Munos. Optimistic planning of deterministic systems. *Recent Advances in Reinforcement Learning*, pages 151–164, 2008.

[23] J.E. Ingersoll. *Theory of Financial Decision Making*. Rowman and Littlefield Publishers, Inc., 1987.

[24] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

[25] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. *Machine Learning: ECML 2006*, pages 282–293, 2006.

[26] J.Z. Kolter and A.Y. Ng. Near-bayesian exploration in polynomial time. In *International Conference on Machine Learning (ICML)*, pages 513–520, 2009.

[27] R. Munos. Optimistic optimization of deterministic functions without the knowledge of its smoothness. In *Neural Information Processing Systems (NIPS)*, 2011.

[28] R. Munos. The optimistic principle applied to games, optimization and planning: Towards Foundations of Monte-Carlo Tree Search. Technical report, 2012.

[29] S.A. Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society, Series B*, 65(2):331–366, 2003.

[30] R. Ortner and P. Auer. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 2007.

[31] J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. In *IEEE-RAS International Conference on Humanoid Robots*, pages 1–20. Citeseer, 2003.

[32] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 697–704, 2006.

[33] M. Riedmiller. Neural fitted Q iteration - first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning (ECML)*, pages 317–328, 2005.

[34] D. Silver and J. Veness. Monte-Carlo planning in large POMDPs. *Neural Information Processing Systems (NIPS)*, 46, 2010.

[35] J. Sorg, S. Singh, and R.L. Lewis. Variance-based rewards for approximate Bayesian reinforcement learning. *Uncertainty in Artificial Intelligence (UAI)*, 2010.

[36] M. Strens. A Bayesian framework for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 943–950, 2000.

[37] R.S. Sutton. Learning to predict by the methods of temporal difference. *Machine Learning*, 3:9–44, 1988.

[38] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. MIT Press, 1998.

[39] T.J. Walsh, S. Goschin, and M.L. Littman. Integrating sample-based planning and model-based reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2010.

[40] A. Weinstein and M.L. Littman. Bandit-based planning and learning in continuous-action Markov decision processes. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2012.