Optimistic planning for belief-augmented Markov decision processes
Raphael Fonteneau, Lucian Busoniu, Rémi Munos

To cite this version:

HAL Id: hal-00840202
https://hal.archives-ouvertes.fr/hal-00840202
Submitted on 1 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 (ε—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 con-
cludes.

II. RELATED OPTIMISTIC APPROACHES

The optimism in the face of uncertainty paradigm has already
lead to several successful results (see [28] for an 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
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 paper). 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 perfor-
ance.

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 $M = (S, A, T, R)$ be a Markov Decision Process
(MDP), where the set $S = \{s^{(1)}, \ldots, s^{(n_S)}\}$ denotes
the finite state space and the set $A = \{a^{(1)}, \ldots, a^{(n_A)}\}$ the
finite action space of the MDP. When the MDP is in state $s_t \in S$ at
time $t \in \mathbb{N}$, an action $a_t \in A$ is selected and the MDP moves
toward a next state $s_{t+1} \in S$ 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 $(s, a, s') \in S \times A \times
S$, which is often true in practice, e.g. in control $R$ is often
known to the user. Let $\pi : S \rightarrow A$ be a deterministic policy,
i.e. a mapping from states to actions. A standard criterion for
evaluating the performance of $\pi$ is to consider its expected
discounted return $J^\pi$ defined as follows:

$$\forall s \in S, \quad J^\pi(s) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1}) | s_0 = s \right]$$
where $\gamma \in [0, 1)$ is the so-called discount factor. An optimal policy is a policy $\pi^*$ such that, for any policy $\pi$,
\[
\forall s \in S, \quad J^\pi(s) \geq J^\pi(s).
\]

Such an optimal policy $\pi^*$ is scored with an optimal return $J^\pi(s) = J^\pi(s)$ which satisfies the Bellman optimality equation:
\[
\forall s \in S, \quad J^\pi(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s') (R(s, a, s') + \gamma J^\pi(s')).
\]

Finding an optimal policy can thus be theoretically achieved by behaving greedily with respect to the optimal state-action value function $Q^* : S \times A \rightarrow \mathbb{R}$ defined as follows:
\[
\forall (s, a) \in S \times A, \quad Q^*(s, a) = \sum_{s' \in S} T(s, a, s') [R(s, a, s') + \gamma J^*(s')].
\]

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 $\mu$. In this setting, an initial prior distribution $b_0$ 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 $b_t$ depends on the prior distribution $b_0$ 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 $b_{t+1}$:
\[
b_{t+1} = P(\mu|h_{t+1}, b_t)
\]
can be updated sequentially:
\[
b_{t+1} = P(\mu|s_t, a_t, s_{t+1}, b_t).
\]

The posterior distribution $b_t$ 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 $z$ obtained by concatenating the state with the belief $z = (s, b)$ 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 $T$ given by:
\[
\forall (z, z') \in \mathbb{B}^2, \forall a \in A, \quad T(z, a, z') = P(z'|z, a) = P(b'|b, s, a, s')E[P(s'|s, a)|b] = 1_{h_{t+1}=(s_t, a, s')}E[P(s'|s, a)|b]
\]

and a reward function $R$ given by:
\[
\forall (z, z') \in \mathbb{B}^2, \forall a \in A, \quad R(z, a, z') = R(s, a, s').
\]

A Bayesian optimal policy $\pi^*$ can be theoretically obtained by behaving greedily with respect to the optimal Bayesian state-action value function $Q^*$:
\[
\forall z \in \mathbb{B}, \quad \pi^*(z) = \arg \max_{a \in A} Q^*(z, a)
\]

where $\forall z \in \mathbb{B}, \forall a \in A$,
\[
Q^*(z, a) = \sum_{z'} T(z, a, z') (R(z, a, z') + \gamma J^*(z')).
\]

Here, $z'$ are reachable belief state when taking action $a$ in belief state $z$ and $J^*(z)$ is the Bayesian optimal return:
\[
J^*(z) = \max_{a \in A} Q^*(z, a).
\]

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 $\pi^*$.

### C. Dirichlet 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 $b$ whose probability density function is:
\[
d(\mu; \Theta) = \prod_{(s, a) \in S \times A} D(\mu_{s, a}; \Theta(s, a, \cdot))
\]

where $D(\cdot; \cdot)$ denotes a Dirichlet distribution, $\Theta(s, a, s')$ denotes the number of observed transitions from $(s, a) \in S \times A$ towards every $s' \in S$ and $\Theta(s, a, \cdot)$ denotes the vector of counters of observed transitions:
\[
\Theta(s, a, s') = \left[ \Theta(s, a, s(1)), \ldots, \Theta(s, a, s(n_s)) \right]
\]

and $\Theta$ is the matrix that contains all $\Theta(s, a, \cdot)$ $s \in S, a \in A$. In the following, we denote by $b(\Theta)$ such a Dirichlet distribution-based posterior. The resulting posterior distribution $b(\Theta)$ satisfies the following well-known property:
\[
E[P(s'|s, a)|b(\Theta)] = \frac{\sum_{s', a, s''} \Theta(s, a, s')}{\sum_{s'', a, s''} \Theta(s, a, s'')}
\]

and the Bayesian update under the observation of a transition $(s, a, s') \in S \times A \times S$ is reduced to a simple increment of the corresponding counter:
\[
\Theta(s, a, s') \leftarrow \Theta(s, a, s') + 1.
\]

In such a context, the Bayesian optimal state-action value function writes:
\[
Q^*(s, b(\Theta), a) = \sum_{s'' \in S} \frac{\Theta(s, a, s')}{\sum_{s'', a, s''} \Theta(s, a, s'')} \left( R(s, a, s') + \gamma J^*(s', b(\Theta')) \right)
\]

where $\Theta'_{s, a, s'}(x, y, x') = \left\{ \begin{array}{ll} \Theta(x, y, x') + 1 & \text{if } (x, y, x') = (s, a, s'), \\ \Theta(x, y, x') & \text{otherwise.} \end{array} \right.$
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 \( x \) and labeled by a BA-state \( z = (s, b(\Theta)) \). Many nodes may have the same label \( z \), for this reason we distinguish nodes from their belief states labels. A node \( x \) is extended by adding to it, for each action \( a \in A \), and then for each \( z' = (s', b(\Theta'_{s,a,s'})) \), a child node \( x' \) labeled by \( z' \). The branching factor of the tree is thus \( n_S \times n_A \). Let us denote by \( C(x, a) \) the set of children \( x' \) corresponding to action \( a \), and by \( C(x) \) the union:

\[
C(x) = \bigcup_{a \in A} C(x, a)
\]

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

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

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

\[
\forall x \in \mathcal{L}(T), \forall a \in A, B(x, a) = \frac{1}{1 - \gamma},
\]

\[
\forall x \in T \setminus \mathcal{L}(T), \forall a \in A, B(x, a) = \sum_{x' \in C(x, a)} T(z, a, z') \left( R(z, a, z') + \gamma \max_{a \in A} B(x', a') \right)
\]

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

![Fig. 1. Illustration of a BA-planning tree. Squares are BA-state nodes whereas circles represent decisions.](image-url)
The difference from the B-values is that its starts with 0 values at the leaves. At the end, the root action $\alpha_n(z_0)$ is selected as follows:

$$\alpha_n(z_0) \in \arg \max_{a'd \in \mathcal{A}} \nu(x_0, a') .$$

Maximizing the lower bound $\nu(x_0, \cdot)$ 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 $n_S \times n_A$, 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 $\Theta$ 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 $n'_S \times n_A$, with $n'_S \ll n_S$.

### V. REINTERPRETATION OF THE BOP ALGORITHM

In this section, we reinterpret 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 $T_h$ of the infinite belief-augmented planning tree $T_\infty$:

$$h : T_H \to \mathcal{A} ,$$

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

$$T_h = \{ x \in T_\infty | x = x_0 \text{ or } \exists x' \in T_h, x \in \mathcal{C}(x', h(x')) \} .$$

where actions $h(x)$ are assigned as desired. The branching factor of $T_h$ is at most $n_S$. Denote the Bayesian expected return of the tree-policy $h$ by $\nu(h)$, and the optimal, maximal Bayesian return by $\nu^*$. A class of tree-policies, $H : T_H \to \mathcal{A}$, is obtained similarly but restricting the procedure to nodes in some finite tree $T_t$ considered by BOP, so that all action assignments below $T_t$ 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 $T_H = T_t \cap T_h$ 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 $z$.

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

$$\nu_H = \sum_{x \in \mathcal{L}(T_H)} P(x)\tilde{R}(x)$$

because the rewards that $h$ can obtain below the leaves of $\mathcal{L}(T_H)$ are lower-bounded by 0. Since rewards are also upper-
bounded by 1, an upper bound on the value of $h \in H$ is:

$$B_H = \sum_{x \in L(T_H)} P(x) \left[ \bar{R}(x) + \frac{\gamma \Delta(x)}{1 - \gamma} \right]$$

$$= \nu_H + \sum_{x \in L(T_H)} c(x) = \nu_H + \text{diam}(H)$$

where we introduce the notations:

$$c(x) = P(x) \frac{\gamma \Delta(x)}{1 - \gamma}$$

the contribution of a leaf $x$ to the difference between the upper and lower bounds, and

$$\text{diam}(H) = \sum_{x \in L(T_H)} c(x)$$

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

$$\delta(h, h') = \sum_{x \in L(T_H \cap T_{h', h})} c(x)$$

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 $T_t$:

$$H^*_t \in \arg \max_{H \in T_t} B_H$$

where $H \in T_t$ means that $T_H \subseteq T_t$. 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 $c(x)$ to the uncertainty $\text{diam}(H^*_t)$ on the value of policies $h \in H^*_t$:

$$x_t \in \arg \max_{x \in L(T_{H^*_t})} c(x)$$

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

$$H^*_n \in \arg \max_{H \in T_n} \nu_H$$

The action $\tilde{a}_n(z_0)$ returned by BOP is then the first action chosen by $H^*_n$.

VI. THEORETICAL RESULTS

Let $R_n(z_0)$ be the Bayesian simple regret:

$$R_n(z_0) = J^*(z_0) - Q^*(z_0, \tilde{a}_n(z_0))$$

i.e. the loss - with respect to the Bayesian optimal policy - of taking action $\tilde{a}_n(z_0)$ instead of $\pi^*(z_0)$. We have the
As the number of observed transitions goes to infinity, the distribution over transition models converges directly to the complexity of planning in a MDP, meaning that the complexity of planning in the BA-MDP becomes similar to the complexity of planning in the lying MDP, where $β$ depends on the belief-augmented state $z$.

#### Proposition 1:

$$\forall z_0 \in B, \lim_{n \to \infty} R_n(z_0) = 0.$$ 

For finite values of $n$, we have:

#### Proposition 2:

$$\forall z_0 \in B, \exists β(z_0) \in \mathbb{R}^+, R_n(z_0) = \begin{cases} O\left(n^{-\frac{1}{\pi^2}}\right) & \text{if } β(z_0) > 0 \\ O\left(\exp\left(-\frac{z_0}{\alpha}\right)^{\frac{1}{2}}\right) & \text{if } β(z_0) = 0 \end{cases}$$

where $β(z_0) \geq 0$ depends on the belief-augmented state $z_0$ and $a, b > 0$ are some constants. Those two propositions directly follow 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). As the number of observed transitions goes to infinity, the distribution over transition models converges towards a Dirac centered on the actual MDP, and the $β(z_0)$ should converges towards the parameter $β(s_0)$ 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 $(n_S = 5)$, and two actions are possible $(n_A = 2)$. Taking action $a^{(1)}$ in state $s^{(1)}$ leads to jump towards state $s^{(1+1)}$, 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.

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 $Θ^{(0)}$ as follows:

$$\forall (s, a, s') \in S \times A \times S, \quad Θ^{(0)}(s, a, s') = 1.$$ 

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 given in Table I with a 95% confidence interval. 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, so it is not surprising that some algorithms may be here more efficient than BOP with $n = 500$, which is likely to be close to Bayesian optimality. We also display in Figure 4 the evolution over time of the empirical probability (computed over the

![Fig. 3. The standard 5-state chain problem.](image)

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEB $(β = 150)$ [26]</td>
<td>165.2</td>
</tr>
<tr>
<td>BEETLE [32]</td>
<td>175.4</td>
</tr>
<tr>
<td>BOP $(n = 50)$</td>
<td><strong>255.6</strong></td>
</tr>
<tr>
<td>BOLT $(n = 150)$ [11]</td>
<td>278.7</td>
</tr>
<tr>
<td>BOLT $(n = 7)$ [1]</td>
<td>289.6</td>
</tr>
<tr>
<td>BOP $(n = 100)$</td>
<td>292.9</td>
</tr>
<tr>
<td>BOSS [2]</td>
<td>300.3</td>
</tr>
<tr>
<td>BOP $(n = 200)$</td>
<td><strong>304.6</strong></td>
</tr>
<tr>
<td>EXPLOIT [32]</td>
<td>307.8</td>
</tr>
<tr>
<td>BOP $(n = 500)$</td>
<td><strong>308.8</strong></td>
</tr>
<tr>
<td>BEB $(β = 1)$ [26]</td>
<td>343.0</td>
</tr>
<tr>
<td>BVR [35]</td>
<td>346.5</td>
</tr>
</tbody>
</table>

Optimal strategy 367.7

**TABLE I**

**PERFORMANCE OF BOP COMPARED WITH OTHER MODEL-BASED BRL APPROACHES.**
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 through interactions within the environment.

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 the meaning of the near-optimality exponent of the belief-augmented MDP and its relationship with the exponent in the underlying MDP.

**Acknowledgments**


**References**


