Planification Optimiste dans les Processus Décisionnels de Markov avec Croyance

Raphael Fonteneau[†,*], Lucian Busoniu[‡], Rémi Munos[*]

[†] Unité de recherche Systèmes et Modélisation, Université de Liège, Belgique
[*] Equipe-projet SequeL, Inria Lille - Nord Europe, France
[‡] Université de Lorraine, CRAN, UMR 7039 et CNRS, CRAN, UMR 7039, France
{raphael.fonteneau, remi.munos}@inria.fr , lucian@busoniu.net

Résumé : Cet article décrit l'algorithme BOP (de l'anglais "Bayesian Optimistic Planning"), un nouvel algorithme d'apprentissage par renforcement Bayésien indirect (c'est à dire fondé sur un modèle). BOP étend l'approche de l'algorithme OP-MDP (de l'anglais "Optimistic Planning for Markov Decision Processes", voir Busoniu *et al.* (2011); Busoniu & Munos (2012)) au cas où les probabilités de transitions du MDP sous-jacent sont initialement inconnues, et doivent être apprises au travers d'interactions avec l'environnement. Les connaissances sur le MDP sous-jacent sont représentées par une distribution de probabilités sur l'ensemble de tous les modèles de transitions à l'aide de distributions de Dirichlet. L'algorithme BOP planifie dans l'espace augmenté état-croyance obtenu par concaténation du vecteur d'état avec la distribution postérieure sur les modèles de transitions. On montre que BOP atteint l'optimalité Bayésienne lorsque le paramètre de budget tend vers l'infini. Quelques expériences préliminaires montrent des résultats encourageants.
Mots-clés : Apprentissage par renforcement Bayésien, optimisme dans l'incertain

1 Introduction

Les algorithmes d'apprentissage pour la planification et la prise de décision sont de plus en plus répendus, notamment dans les domaines de la finance (Ingersoll (1987)), de la robotique (Peters *et al.* (2003); Riedmiller (2005)), de l'intelligence artificielle (Sutton & Barto (1998)) ou même de la médecine (Murphy (2003)). En collectant progressivement des informations sur leur environnement, ces algorithmes sont en mesure d'apprendre un comportement optimal par rapport à un critère donné a priori.

De nombreux défis sont à relever afin de mettre au point de tels algorithmes efficaces. En particulier, une difficulté majeure consiste à résoudre le compromis Exploration / Exploitation (E/E) : à chaque instant, l'algorithme doit choisir entre (i) prendre une décision de bonne qualité par rapport aux connaissances acquises par le passé ("Exploitation") et (ii) prendre des décisions susceptibles de mener à l'acquisition de nouvelles informations afin d'être en mesure de prendre de meilleures décisions dans le futur ("Exploration"). Ce problème intrigue les chercheurs depuis des décennies ; dans les années 60, la communauté scientifique travaillant dans le domaine du contrôle optimal développait déjà la théorie du Control Dual (le terme "dual" référant explicitement au double objectif E/E, Feldbaum (1960)), montrant en particulier qu'un tel compromis pouvait être théoriquement résolu en utilisant la programmation dynamique (Bellman (1957)).

A la fin des années 80, la popularisation de l'apprentissage par renforcement (RL) (Sutton (1988)) a donné un nouvel élan à la communauté travaillant sur la mise au point d'algorithmes d'apprentissage dans les environements incertains. Le compromis E/E a ainsi été redécouvert à la lumière du paradigme RL. Dans un premier temps, des approches heuristiques ont été proposées ($\epsilon-$greedy, exploration Boltzmann), mais plus tard, à la fin de années 90, de nouvelles perspectives ont été ouvertes grâce à l'apport de techniques Bayésiennes, donnant naissance à l'apprentissage par renforcement Bayésien (BRL, Dearden *et al.* (1998); Strens (2000)). L'atout principal du BRL est de formaliser de manière élégante le compromis E/E et d'en définir une solution théorique. Cependant, en pratique, les approches BRL se sont montrées très gourmandes en temps de calcul, voire

quasi-insolubles, à l'exception des problèmes de bandits à k bras où l'approche Bayésienne mène aux fameux indices de Gittins (Gittins (1989)). En dépit de ce défi calculatoire, le BRL est dévenu de plus en plus populaire au cours de la dernière décennie (Poupart *et al.* (2006)), bien que les algorithmes BRL n'atteignaient pas encore les performances des algorithmes RL classiques (voir par exemple Brafman & Tennenholtz (2003)).

Plus récemment, une nouvelle génération d'algorithmes basés sur des techniques de développent d'arbres a offert de belles avancées en terme de performance empiriques. En particulier, les techniques de type MCTS (de l'anglais "Monte Carlo Tree Search", voir Coulom (2007); Munos (2012)), et notamment l'algorithme UCT (de l'anglais "Upper Confidence Trees", Kocsis & Szepesvári (2006)) ont permis d'aborder des problèmes à grande échelle tels que le jeu de Go (Gelly *et al.* (2006)). De telles techniques sont actuellement en cours d'importation dans le BRL, donnant naissance à de nouveaux algorithmes performants (Silver & Veness (2010); Asmuth & Littman (2011); Guez *et al.* (2012)).

La contribution décrite dans ce papier se situe dans le contexte présenté ci-dessus, à la croisée du BRL indirect (c'est-à-dire fondé sur un modèle) et des algorithmes à base de développement d'arbres. On décrit l'algorithme BOP (de l'anglais "Bayesian Optimistic Planning"), un nouvel algorithme BRL indirect initialement proposé par Fonteneau *et al.* (2013). BOP étend le principe de l'algorithme OP-MDP (de l'anglais "Optimistic Planning for MDPs", voir Busoniu *et al.* (2011); Busoniu & Munos (2012)) au cas où la dynamique de l'environnement est initialement inconnue et doit être apprise par intéraction avec l'environement. Le principe de l'optimisme dans l'incertain utilisé par OP-MDP est étendu au cas BA-MDP (de l'anglais "Belief-Augmented MDP", voir Dimitrakakis & Lagoudakis (2008)) obtenu par concaténation du vecteur d'état avec la distribution postérieure sur l'espace des modèles de transitions possibles. L'algorithme BOP construit un BA-arbre de planification dans l'espace des BA-états en partant du BA-état courant. Itérativement, de nouveaux noeuds du BA-arbre sont développés, en ajoutant, pour chaque action, tous les BA-états successeurs possibles. Le nombre maximal d'ouvertures de noeuds est limité à un budget n afin de limiter les temps de calcul. Le principle de l'optimisme dans l'incertain est mis à profit afin de partager efficacement le budget de n noeuds et développer en priorité les noeuds les plus prometteurs. Cette approche est rendue possible en particulier grâce à l'utilisation de distributions de Dirichlet pour chaque pair état-action, ce qui contraint le facteur de branchement des arbres, se trouvant être le même que dans le contexte OP-MDP. De manière analogue à OP-MDP, BOP peut être réinterprété comme une technique d'optimisation de type séparation et évaluation (en anglais, "branch and bound") dans un espace de politiques. Les résultats théoriques associés à l'algorithme OP-MDP s'appliquent également dans le contexte BA-MPD, montrant ainsi que BOP tend à prendre des actions optimales (au sens Bayésien) lorsque le budget n tend vers l'infini. Quelques illustrations expérimentales sont obtenues sur le problème chaîne à 5 états (Strens (2000)).

La suite de cet article s'articule de la manière suivante : dans la section 2, des travaux connexes relatifs à l'utilisation du principe de l'optimisme dans l'incertain pour résoudre des MDPs sont discutés. La section 3 formalise le problème du BRL indirect considéré dans cet article. La section 4 décrit la contribution principale de cet article, l'algorithme BOP. En section 5, BOP est réinterprété comme une technique d'optimisation de type séparation et évaluation, tandis que la convergence vers l'optimalité Bayésienne est donnée en section 6. La section 7 présente quelques résultats de simulation, puis la section 8 conclut en ouvrant quelques perspectives.

2 Travaux connexes

Le paradigme de *l'optimisme dans l'incertain* est à la source de nombreux succès (voir Munos (2012)), en particulier dans les domaines des bandits multi-bras (Auer *et al.* (2002); Bubeck *et al.* (2009)), de la planification pour les systèmes déterministes (Hren & Munos (2008)) ou stochastiques (Kocsis & Szepesvári (2006); Walsh *et al.* (2010); Bubeck & Munos (2010); Asmuth & Littman (2011); Busoniu *et al.* (2011); Busoniu & Munos (2012); Weinstein & Littman (2012)) quand la dynamique du système est connu, et aussi pour les problèmes d'optimisation de fonctions uniquement accessibles via l'échantillonnage (Munos (2011)).

Le principle de l'optimisme dans l'incertain a également été utilisé pour aborder le dilemme E/E dans le contexte des MDP lorsque le modèle de transition est initialement inconnu et appris au

travers d'interactions avec l'environnement. Par exemple, l'algorithme R-MAX (Brafman & Tennenholtz (2003)) accorde des récompenses optimistes aux transitions moins visitées. L'algorithme UCRL / UCRL2 (Ortner & Auer (2007); Jaksch *et al.* (2010)) agit aussi de manière optimiste en utilisant des bornes UCB. Très récemment, (Castronovo *et al.* (2012)) ont proposé de résoudre le dilemme E/E dans le contexte où il est possible de tirer des MDP à partir d'une distribution fixée a priori (et non mise à jour à partir des échantillons observés). Un algorithme de bandit multi-bras est par la suite utilisé afin d'identifier des politiques efficaces dans un espace de politiques basées sur des formules, chaque politique étant associée à un bras.

Le principe de l'optimisme dans l'incertain a également été proposé dans le contexte BRL. L'algorithme BEB (de l'anglais "Bayesian Exploration Bonus", voir Kolter & Ng (2009)) est une approche indirecte (basée sur un modèle) visant à prendre des décisions étant donné le modèle espéré courant pour lequel un bonus est accordé aux paires état-action ayant été mois observées. L'idée d'ajouter un tel bonus est également mise à profit par l'algorithme BVR (de l'anglais "Bounded Variance Reward", voir Sorg *et al.* (2010)) en utilisant des bonus différents. L'algorithme BOSS (de l'anglais "Best Of Sampled Set", voir Asmuth *et al.* (2009)) propose une approche de type Thompson en (i) échantillonnant des modèles à partir d'une distribution postérieure et (ii) en combinant ces modèles sous la forme d'un MDP optimiste afin de prendre des décisions. Une variante de l'algorithme BOSS utilisant un échantillonnage adaptatif a également été proposée par Castro & Precup (2010). Plus récemment, l'algorithme BOLT (de l'anglais "Bayesian Optimistic Local Transitions", voir Araya *et al.* (2012)) adopte également une approche optimiste en suivant une politique optimiste par rapport à une variante optimiste du modèle espéré courant (obtenu en ajoutant des transitions optimistes aléatoires). Encore plus récemment, l'algorithme BAMCP (de l'anglais "Bayes-Adaptive Monte Carlo Planning", voir Guez *et al.* (2012)) propose une approche de type UCT combinée à un échantillonnage parcimonieux donnant de bons résultats théoriques et empiriques.

Comme toutes les méthodes listées ci-dessus, l'algorithme BOP se place dans la classe des méthodes utilisant le principe de l'optimisme dans l'incertain, dans le contexte BRL indirect. BOP a la particularité de mettre à jour la croyance pendant la phase de planification, ce qui assure que, indépendamment des transitions observées, BOP converge vers l'optimum Bayésien lorsque que le paramètre de budget n tend vers l'infini.

3 Formalisation du problème

Le problème standard de l'apprentissage par renforcement (RL) est formalisé en section 3.1. En section 3.2, on formalise le problème du RL Bayésien indirect que l'on spécifie avec des distributions de Dirichlet en Section 3.3.

3.1 Apprentissage par renforcement

Soit $M = (\mathcal{S}, \mathcal{A}, T, R)$ un processus de décision de Markov (MDP). L'ensemble $\mathcal{S} = \left\{s^{(1)}, \ldots, s^{(n_\mathcal{S})}\right\}$ désigne l'espace d'état fini et l'ensemble $\mathcal{A} = \left\{a^{(1)}, \ldots, a^{(n_\mathcal{A})}\right\}$ l'espace de décision (ou action) fini du MDP. Lorsque le MDP se trouve dans un état $s_t \in \mathcal{S}$ au temps $t \in \mathbb{N}$, une décision $a_t \in \mathcal{A}$ est prise et le MDP fait une transition vers un nouvel état $s_{t+1} \in \mathcal{S}$ tiré selon une distribution de probabilité

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

Une récompense instantanée scalaire $r_t \in [0, 1]$ est alors reçue :

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

Dans cet article, on fait l'hypothèse que le modèle de transitions T est inconnue. Par soucis de simplicité, on fait également l'hypothèse que la fonction de récompense instantanée $R(s, a, s') \in [0, 1]$ est connue pour chacune des transitions possibles $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, ce qui est souvent le cas en pratique. Soit $\pi : \mathcal{S} \to \mathcal{A}$ une politique de décision déterministe, c'est-à-dire une fonction de l'espace d'état vers l'espace de décision. Un critère standard pour évaluer les performances de

la politique π est l'espérance de la somme des récompenses actualisées J^π :

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

où $\gamma \in [0, 1)$ est le facteur d'actualisation. Une politique de décision optimale π^* est telle que, pour toute politique π,

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

Une politique optimale π^* possède un retour optimal $J^*(s) = J^{\pi^*}(s)$ qui satisfait l'équation de Bellman :

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

Un telle politique optimale s'obtient en agissant de manière gloutonne par rapport à la fonction de valeur optimale Q^* :

$$\forall s \in \mathcal{S}, \quad \pi^*(s) = \arg\max_{a \in \mathcal{A}} \quad Q^*(s, a)$$

où $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ définie ainsi :

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

Ici, la difficulté principale réside dans le fait que le modèle de transition $T(\cdot, \cdot, \cdot)$ est initialement inconnu, et progressivement découvert par intéractions. Cela implique de faire un compromis entre (i) agir de manière optimale par rapport aux connaissances disponibles sur le modèle (exploitation) et (ii) prendre des décisions susceptibles de mener à l'acquisition de nouvelles informations sur le modèle de transitions, permettant la prise de meilleures décision à plus long terme (exploration).

3.2 Apprentissage par renforcement Bayésien indirect

L'apprentissage par renforcement indirect (fondé sur un modèle) aborde le dilemme exploration / exploitation (E/E) en formalisant la connaissance sur le modèle de transition inconnu au moyen d'une distribution de probabilité sur l'ensemble de tous les modèles de transition possibles $\boldsymbol{\mu}$. Une distribution initiale $\boldsymbol{b}_0$ est donnée (la distribution prior) et séquentiellement mise à jour selon la règle de Bayes afin de prendre en compte les nouveaux échantillons observés sur le modèle inconnu. A chaque pas de temps t, la distribution postérieure $\boldsymbol{b}_t$ dépend du prior $\boldsymbol{b}_0$ et de l'historique $h_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$ observé jusque là. L'hypothèse Markovienne implique que la distribution postérieure $\boldsymbol{b}_{t+1}$:

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

peut être mise à jour de façon séquentielle :

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

La distribution postérieure $\boldsymbol{b}_t$ est courramment appelée "croyance" dans la littérature BRL.

Une approache standard afin de − théoriquement − résoudre les problèmes de BRL consiste à considérer un BA-état $\boldsymbol{z}$ obtenu par concaténation de l'état avec la croyance $\boldsymbol{z} = \langle s, \boldsymbol{b} \rangle$ et de résoudre le BA-MDP correspondant (Duff (2002); Dimitrakakis (2008)). Dans la suite, on désigne par $\mathbb{B}$ l'espace de BA-état. Ce BA-MDP est caractérisé par une modèle de transitions $\mathbf{T}$:

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

et une fonction de récompense $\mathbf{R}$:

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

Dans un tel contexte, une politique optimale Bayésienne $\boldsymbol{\pi}^*$ peut être théoriquement obtenue en agissant de façon gloutonne par rapport à la fonction de valeur optimal Bayésienne $\mathbf{Q}^*$:

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

où

$$\forall \boldsymbol{z} \in \mathbb{B}, \forall a \in \mathcal{A}, \quad \mathbf{Q}^*(\boldsymbol{z}, a) = \sum_{\boldsymbol{z}'} \mathbf{T}\left(\boldsymbol{z}, a, \boldsymbol{z}'\right) \left(\mathbf{R}(\boldsymbol{z}, a, \boldsymbol{z}') + \gamma \mathbf{J}^*(\boldsymbol{z}')\right) \ .$$

Ici, $\boldsymbol{z}'$ est un BA-état accessible en prenant l'action a dans le BA-état $\boldsymbol{z}$ et $\mathbf{J}^*(\boldsymbol{z})$ est le retour optimal Bayésien :

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

Dans cet article, l'objectif est d'approcher $\boldsymbol{\pi}^*$ le plus précisément possible.

3.3 Distributions de Dirichlet

Une approche courramment utilisée en BRL est d'utiliser des distributions de Dirichlet indépendantes pour chaque couple état-action. On obtient une distribution postérieure dont la fonction de densité s'exprime :

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

où $D(\cdot;\cdot)$ désigne une distribution de Dirichlet, $\boldsymbol{\Theta}(s, a, s')$ désigne le nombre total de transitions observées depuis $(s, a) \in \mathcal{S} \times \mathcal{A}$ vers chaque $s' \in \mathcal{S}$, $\boldsymbol{\Theta}(s, a, \cdot)$ désigne le vecteur de compteurs des transitions observées :

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

et $\boldsymbol{\Theta}$ est le tenseur contenant tous les $\boldsymbol{\Theta}(s, a, .), s \in \mathcal{S}, a \in \mathcal{A}$. On désigne par $\boldsymbol{b}\left(\boldsymbol{\Theta}\right)$ la distribution postérieure fondée sur les distributions de Dirichlet ainsi obtenue. Une telle distribution $\boldsymbol{b}\left(\boldsymbol{\Theta}\right)$ satisfait la propriété suivante :

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

et la mise à jour selon la règle de Bayes suite à l'observation de la transition $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ est réduite à l'incrémentation du compteur correspondant :

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

Dans un tel contexte, la fonction de valeur optimale Bayésienne s'écrit :

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

où $\boldsymbol{\Theta}'_{s,a,s'}$ est tel que :

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

4 L'algorithme BOP

Cette section décrit la contribution principale de cet article, l'algorithme BOP (de l'Anglais *Bayesian Optimistic Planning*), initialement proposé par Fonteneau *et al.* (2013). On formalise d'abord la notion de *BA-arbre de planification* en section 4.1. L'algorithme BOP se fonde sur une approche optimiste de développent de BA-arbres de planification que l'on détaille en section 4.2.

4.1 BA-arbre de planification

Chaque noeud d'un BA-arbre de planification est noté $\boldsymbol{x}$ et étiqueté par un BA-état $\boldsymbol{z} = \langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$. Plusieurs noeuds peuvent avoir la même étiquette $\boldsymbol{z}$, c'est pour cette raison qu'on fait la distinction entre les noeuds et leurs étiquettes. Un noeud $\boldsymbol{x}$ est ouvert (ou développé) lorsqu'on lui adjoint, pour chaque action $a \in \mathcal{A}$, et pour chaque BA-état successeur $\boldsymbol{z}' = \left\langle s', \boldsymbol{b}\left(\boldsymbol{\Theta}'_{s,a,s'}\right)\right\rangle$, un noeud fils $\boldsymbol{x}'$ étiqueté par $\boldsymbol{z}'$. Le facteur de branchement de l'arbre est ainsi $n_{\mathcal{S}} \times n_{\mathcal{A}}$. Soit $\mathcal{C}(\boldsymbol{x}, a)$ l'ensemble de tous les noeuds fils $\boldsymbol{x}'$ correspondant à l'action a, et soit $\mathcal{C}(\boldsymbol{x})$ l'ensemble :

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

On introduit quelques notations :
- Le BA-arbre est noté $\mathcal{T}$, son ensemble de feuilles $\mathcal{L}(\mathcal{T})$;
- Un noeud de l'arbre $\boldsymbol{x}$ est étiqueté par son BA-état $\boldsymbol{z} = \langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$. Un noeud fils est noté $\boldsymbol{x}'$ (et étiqueté $\boldsymbol{z}' = \left\langle s', \boldsymbol{b}\left(\boldsymbol{\Theta}'_{s,a,s'}\right)\right\rangle$ où a est l'action prise pour transiter de $\boldsymbol{z}$ à $\boldsymbol{z}'$).
- La profondeur d'un noeud $\boldsymbol{x}$ est notée $\Delta(\boldsymbol{x})$.

Un BA-arbre est représenté en figure 1.

4.2 Planification optimiste dans un espace de BA-états

L'algorithme BOP construit un BA-arbre de planification à partir d'un noeud racine étiqueté par le BA-état pour lequel une action doit être prise. A chaque pas de temps, l'algorithme choisit une feuille de l'arbre et l'ouvre en lui ajoutant, pour chaque action, tous les BA-états successeurs possibles. Cette procédure d'ouverture de feuilles se termine lorsqu'un budget d'ouvertures fixé $n \in \mathbb{N} \setminus \{0\}$ est atteint, et une action est choisie en fonction de l'arbre ainsi développé. La clé de cette stratégie se trouve dans la procédure de sélection des feuilles à ouvrir. On met en place une stratégie optimiste pour choisir quelles feuilles ouvrir en priorité.

Critère d'ouverture. Pour chaque noeud $\boldsymbol{x} \in \mathcal{T}$ (étiqueté par $\boldsymbol{z} = \langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle$) et chaque décision $a \in \mathcal{A}$, on définit récursivement la B-valeur $B(\boldsymbol{x}, a)$ de la manière suivante :

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

Chaque B-valeur $B(\boldsymbol{x}, a)$ est une borne supérieure sur la valeur de la fonction de valeur optimale Bayésienne $\mathbf{Q}^*(\langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle, a)$.

Afin d'obtenir un ensemble de feuilles candidates à l'ouverture, on construit un sous-arbre optimiste à partir de la racine en sélectionnant à chaque noeud les fils associés à des actions optimistes :

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

(les indéterminées sont levées de façon déterministe). On désigne par $\mathcal{T}^{\dagger}$ le sous-BA-arbre optimiste résultant, et par $\mathcal{L}\left(\mathcal{T}^{\dagger}\right)$ son ensemble de feuilles. Un sous-abre optimiste est représenté à la Figure 2.

Pour choisir une feuille parmis les candidates $\mathcal{L}\left(\mathcal{T}^{\dagger}\right)$, on propose de maximiser la réduction potentielle de la B-valeur à la racine du BA-arbre $B\left(\boldsymbol{x}_0, a^{\dagger}(\boldsymbol{x}_0)\right)$. La B-valeur peut en effet être

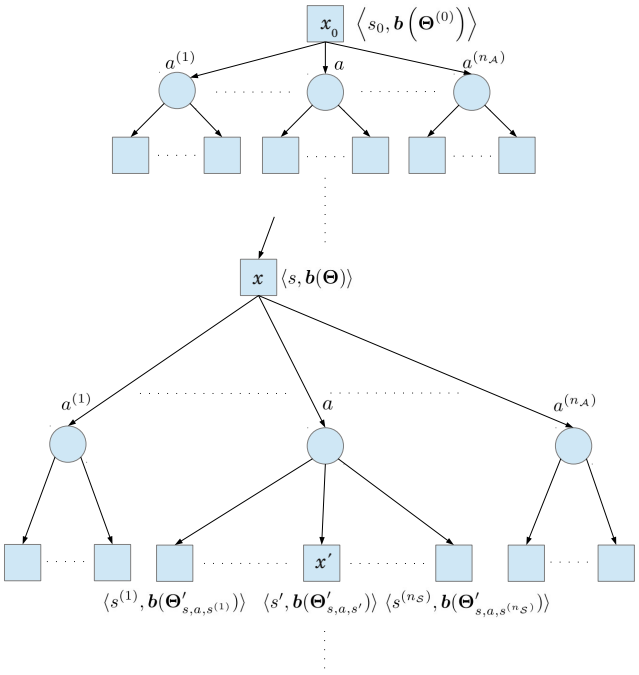

FIGURE 1 – Illustration d'un BA-arbre. Les carrés sont les BA-états tandis que les cercles représentent les décisions.

écrite sous la forme d'un retour optimiste espéré obtenu le long des chemins depuis la racine vers les feuilles du sous-arbre optimiste :

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

où $\mathbf{P}(\boldsymbol{x})$ est la probabilité d'atteindre $\boldsymbol{x} \in \mathcal{L}\left(\mathcal{T}^\dagger\right)$ (produit des probabilités le long du chemin) et $\bar{\mathbf{R}}(\boldsymbol{x})$ est la somme des récompenses actualisées obtenues sur le chemin. En désignant le chemin par $\boldsymbol{y}^{\boldsymbol{x}}_0, \boldsymbol{y}^{\boldsymbol{x}}_1, \ldots, \boldsymbol{y}^{\boldsymbol{x}}_{\Delta(\boldsymbol{x})}$ pour un certain $\boldsymbol{x}$ et $\boldsymbol{z}^{\boldsymbol{x}}_0, \boldsymbol{z}^{\boldsymbol{x}}_1, \ldots, \boldsymbol{z}^{\boldsymbol{x}}_{\Delta(\boldsymbol{x})}$ la séquences d'étiquettes associées ($\boldsymbol{y}^{\boldsymbol{x}}_0 = \boldsymbol{x}_0$ et $\boldsymbol{y}^{\boldsymbol{x}}_{\Delta(\boldsymbol{x})} = \boldsymbol{x}$), on a :

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

($\mathbf{P}$ et $\bar{\mathbf{R}}$ sont définies sur des noeuds). La contribution d'une feuille dans l'équation 1 s'exprime :

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

Si une telle feuille était ouverte, sa contribution diminuerait le plus si les récompenses sur toutes les transitions vers les noeuds fils étaient nulles. Dans un tel cas, la contribution mise à jour serait $\mathbf{P}(\boldsymbol{x}) \left(\bar{\mathbf{R}}(\boldsymbol{x}) + \frac{\gamma^{\Delta(\boldsymbol{x})+1}}{1-\gamma}\right)$, et sa contribution aurait donc diminué de :

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

Finalement, la règle pour choisir une feuille à ouvrir $\boldsymbol{x}_e$ est la suivante :

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

Algorithm 1 L'algorithme BOP.

input BA-état initial $\boldsymbol{z}_0 = \left\langle s_0, \boldsymbol{b}\left(\boldsymbol{\Theta}^{(0)}\right)\right\rangle$; budget n ;
output action quasi-optimale Bayésienne $\tilde{a}_n(\boldsymbol{z}_0)$;
initialization $\mathcal{T}_0 \leftarrow \{\boldsymbol{x}_0\}$;
for $t = 0, \ldots, n-1$ **do**
  à partir de $\boldsymbol{x}_0$, construire le sous-BA-arbre optimiste $\mathcal{T}_t^\dagger$;
  sélectionner une feuille à ouvrir : $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}\in\mathcal{L}(\mathcal{T}_t^\dagger)} \mathbf{P}(\boldsymbol{z})\gamma^{\Delta(\boldsymbol{x})}$;
  ouvrir $\boldsymbol{x}_t$ afin d'obtenir $\mathcal{T}_{t+1}$;
end for
return $\tilde{a}_n(\boldsymbol{z}_0) \in \arg\max_{a\in\mathcal{A}} \nu(\boldsymbol{x}_0, a)$; appliquer l'action $\tilde{a}_n(\boldsymbol{z}_0)$; observer l'état suivant $\tilde{s}$;
Bayes update $\boldsymbol{\Theta}^{(0)}(s_0, \tilde{a}_n(\boldsymbol{z}_0), \tilde{s}) \leftarrow \boldsymbol{\Theta}^{(0)}(s_0, \tilde{a}_n(\boldsymbol{z}_0), \tilde{s}) + 1$

Choix de l'action à la racine. De manière analogue aux B-valeurs, on définit les ν-valeurs :

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

L'unique différence avec les B-valeurs est qu'on initialise avec des valeurs 0 dans les feuilles. Similairement aux B-valeurs, chaque $\nu-$valeur $\nu(\boldsymbol{x}, a)$ est une borne inférieure sur la fonction de valeur optimale Bayésienne $\mathbf{Q}^*(\langle s, \boldsymbol{b}(\boldsymbol{\Theta})\rangle, a)$.

Finalement, l'action choisie $\tilde{a}_n(\boldsymbol{z}_0)$ est telle que :

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

Maximiser la borne inférieure $\nu(\boldsymbol{x}_0, \cdot)$ peut être interprété comme le fait de prendre une décision prudente. On donne en table 1 un pseudo-code de l'algorithme BOP.

Notons que le facteur de branchement du BA-arbre est $n_{\mathcal{S}} \times n_{\mathcal{A}}$, c'est-à-dire le même que le facteur de branchement dans le cas des arbres de planification utilisés par l'algorithme OP-MDP. L'ajout de complexité apporté par BOP est la propagation et la mise à jour des compteurs $\boldsymbol{\Theta}$ dans le BA-arbre, afin de mettre à jour les probabilité selon la règle de Bayes. Notons également que dans le cas d'applications réelles, on constate souvent que l'ensemble des états atteignables à partir d'un état donné est beaucoup plus petit que $\mathcal{S}$. Lorsqu'une telle connaissance est disponible, elle peut être mise à profit par BOP, et le facteur de branchement devient $n'_{\mathcal{S}} \times n_{\mathcal{A}}$ avec $n'_{\mathcal{S}} \ll n_{\mathcal{S}}$.

5 Réinterpretation de l'algorithme BOP

Dans cette partie, on propose une réinterprétation de l'algorithme BOP comme une technique d'optimisation de type séparation et évaluation (en anglais, *branch and bound*) dans l'ensemble des *politiques BA-arbres*. Une politique BA-arbre h est une affectation d'actions à un sous-abre $\mathcal{T}_h$ du BA-arbre infini $\mathcal{T}_\infty$:

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

en prenant récursivement en compte uniquement les noeuds atteints après les choix d'actions faits jusque là :

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

où les actions $h(\boldsymbol{x})$ sont choisies comme voulues. Le facteur de branchement de $\mathcal{T}_h$ vaut au plus $n_{\mathcal{S}}$. On désigne par $\mathbf{v}(h)$ le retour Bayésien espéré de la politique BA-arbre h.

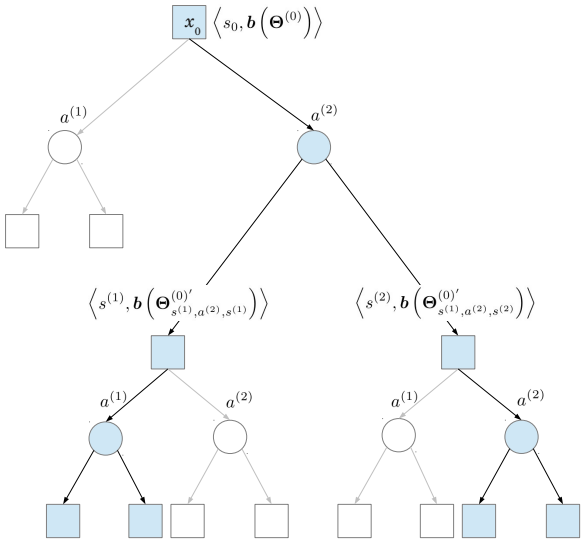

FIGURE 2 – Un sous-BA-arbre optimiste dans le cas $n_{\mathcal{S}} = n_{\mathcal{A}} = 2$. Les morceaux de l'arbre original n'appartenant pas au sous-BA-arbre optimiste sous en gris-clair/blanc.

On obtient une classe de politiques BA-arbre, $H : \mathcal{T}_H \to \mathcal{A}$, en restreignant la procédure à un arbre fini $\mathcal{T}_t$ considéré par BOP, en laissant toutes les actions au delà de l'arbre $\mathcal{T}_t$ non affectées. H est un ensemble de politiques BA-arbres, dans lequel une politique $h \in H$ est entièrement spécifiée dès lors que les actions initialement laissées libres sont fixées. Notons que $\mathcal{T}_H = \mathcal{T}_t \cap \mathcal{T}_h$ pour tout $h \in H$. Notons que les politiques BA-arbre ne sont pas standards, dans le sens où on utilise plus fréquemment des politiques ne dépendant que de l'état courant, qui sont suffisantes pour atteindre l'optimalité. L'utilisation de politique BA-arbre permet de mieux illustrer notre approche, sans perdre en terme de généralité.

Le retour Bayésien espéré d'une politique BA-arbre h appartenant à la classe H est borné inférieurement par :

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

car les récompenses que h peut obtenir sous les feuilles de $\mathcal{L}(\mathcal{T}_H)$ sont bornées par 0. Puisque les récompenses sont également bornées par 1, une borne supérieure sur la valeur de $h \in H$ est :

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

où on utilise la notation :

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

pour désigner la contribution d'une feuille $\boldsymbol{x}$ à l'écart entre la borne supérieure et la borne inférieure, et

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

le diamètre de H. Notons que $\mathrm{diam}(H) = \sup_{h,h' \in H} \delta(h, h')$ où δ est une métrique définie sur l'espace des politiques BA-arbre :

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

En utilisant ces notations, BOP peut être reformulé de la façon suivante. A chaque itération, l'algorithme choisit une classe de politiques BA-arbre optimistes qui maximise la borne supérieure parmis toutes les classes compatibles avec l'arbre courant $\mathcal{T}_t$:

$$H_t^\dagger \in \arg\max_{H \in \mathcal{T}_t} \quad B_H$$

où $H \in \mathcal{T}_t$ signifie que $\mathcal{T}_H \subseteq \mathcal{T}_t$. La classe de politiques BA-arbre optimistes est explorée plus profondément, en ouvrant une des feuilles (et figeant le choix d'action menant à ce noeud). La feuille choisie est celle maximisant la contribution $c(\boldsymbol{x})$ à l'incertitude diam $\left(H_t^\dagger\right)$ sur la valeur des politiques BA-arbre $h \in H_t^\dagger$:

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

En utilisant la métrique δ, on peut également voir ce procédé comme une séparation de l'ensemble des politiques BA-arbre H selon la plus longue arrête, où H est un hyperectangle à $\left|\mathcal{L}\left(\mathcal{T}_{H_t^\dagger}\right)\right|$ dimensions, de longueur $c(\boldsymbol{x})$ selon la dimension $\boldsymbol{x}$. L'algorithme poursuit à l'itération suivante avec le nouveau BA-arbre $\mathcal{T}_{t+1}$. Après n itérations, une classe de politiques BA-arbre est choisie, celle maximisant la borne inférieure :

$$H_n^* \in \arg\max_{H \in \mathcal{T}_n} \quad \nu_H \ .$$

L'action $\tilde{a}_n(\boldsymbol{z}_0)$ retournée par BOP est la première action de H_n^*.

6 Propriétés théoriques

On définit le regret simple Bayésien $\mathcal{R}_n(\boldsymbol{z}_0)$:

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

c'est-à-dire le coup lié à la prise de l'action $\tilde{a}_n(\boldsymbol{z}_0))$ à la place de la politique Bayésienne optimale $\boldsymbol{\pi}^*(\boldsymbol{z}_0)$. On a le résultat suivant :

Theorème : Pour tout BA-état $\boldsymbol{z}_0 \in \mathbb{B}$, il existe un *exposant de quasi-optimalité* $\beta(\boldsymbol{z}_0) \in \left[0, \frac{\log(n_{\mathcal{A}} n_{\mathcal{S}})}{\log(1/\gamma)}\right]$ tel que :

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

et lorsque $\beta(\boldsymbol{z}_0) = 0$, le regret décroit exponentiellement avec n :

$$\exists a, b > 0 : \mathcal{R}_n(\boldsymbol{z}_0) = O\left(\exp\left(-\left(\frac{n}{a}\right)^{\frac{1}{b}}\right)\right) \text{ si } \beta(\boldsymbol{z}_0) = 0.$$

S'ensuit alors le résultat :

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

Ce résultat découle directement de l'analyse de l'algorithme OP-MDP (voir Busoniu & Munos (2012), matériel supplémentaire), que l'on applique ici dans le contexte d'un BA-MDP (qui est aussi un MDP). L'exposant de quasi-optimalité $\beta(\boldsymbol{z}_0)$ mesure le taux d'accroissement d'un certain ensemble de noeuds importants dans le BA-arbre enraciné en $\boldsymbol{z}_0$: intuitivement, les noeuds ayant une large contribution aux politiques quasi-optimales au sens Bayésien. $\beta(\cdot)$ varie entre 0 (problème facile) et $\log\left(n_{\mathcal{A}} n_{\mathcal{S}}\right) / \log\left(1/\gamma\right)$ (problème difficile).

Lorsque le nombre de transitions observées tend vers l'infini, et sous certaines hypothèses concernant le prior, la distribution postérieure sur les modèles de transition converge vers un Dirac centré sur le vrai MDP. On conjecture alors que $\beta(\boldsymbol{z}_0)$ converge également vers le paramètre $\beta(s_0)$ du MDP sous-jacent, signifiant intuitivement que la complexité du problème de planification dans le BA-MDP devient alors similaire à la planification dans le "vrai" MDP.

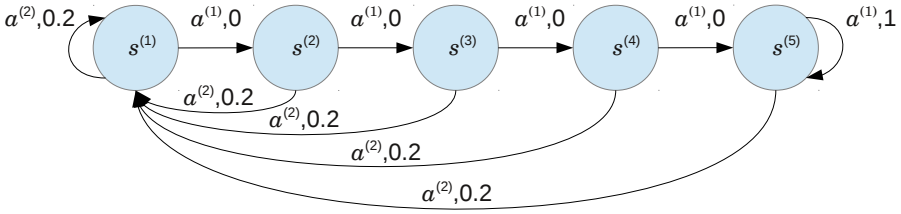

FIGURE 3 – Le problème chaîne à 5 états.

Algorithme	Performance
BEB ($\beta = 150$) [Kolter & Ng (2009)]	165.2
BEETLE [Poupart *et al.* (2006)]	175.4
BOP ($n = 50$)	**255.6**
BOLT ($\eta = 150$) [Araya *et al.* (2012)]	278.7
BOLT ($\eta = 7$) [Araya *et al.* (2012)]	289.6
BOP ($n = 100$)	**292.9**
BOSS [Asmuth *et al.* (2009)]	300.3
BOP ($n = 200$)	**304.6**
EXPLOIT [Poupart *et al.* (2006)]	307.8
BOP ($n = 500$)	**308.8**
BEB ($\beta = 1$) [Kolter & Ng (2009)]	343.0
BVR [Sorg *et al.* (2010)]	346.5
Stratégie optimale	**367.7**

TABLE 1 – Performances de BOP comparées à d'autres techniques BRL indirectes sur le problème chaîne à 5 états avec prior uniforme.

7 Illustration

On compare l'algorithme BOP avec d'autre approches BRL indirectes sur le problème chaîne à 5 états (Strens (2000)), qui est un problème-jouet classiquement utilisé pour comparer les algorithmes BRL. L'espace d'état contient 5 états ($n_{\mathcal{S}} = 5$), et deux actions sont possibles ($n_{\mathcal{A}} = 2$). Prendre l'action $a^{(1)}$ dans l'état $s^{(i)}$ provoque la transition vers l'état $s^{(i+1)}$, sauf dans l'état $s^{(5)}$ où cela fait demeurer dans l'état $s^{(5)}$ tout en recevant une récompense de $+1$. Prendre l'action $a^{(2)}$ fait revenir (ou rester) dans l'état $s^{(1)}$ tout en recevant une récompense de .2. Avec une probabilité $p = .2$, prendre une action a l'effet de l'autre action. La stratégie optimale consiste à prendre l'action $a^{(1)}$ indépendamment de l'état. Une illustration est donnée en figure 3.

Le modèle de transition est inconnu. Dans nos expériences, on considère une distribution prior uniforme, ce qui signifie qu'on n'incorpore aucune connaissance a priori (toutes les transitions sont donc envisageables). Dans le cas particulier des distributions de Dirichlet, cette hypothèse est implémentée en initialisant les compteurs $\mathbf{\Theta}^{(0)}$ comme suit :

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

L'algorithme BOP a été testé 500 fois à partir de l'état $s_0 = s^{(1)}$, où les actions suggérées par BOP ont été appliquées pendant 1000 pas de temps, ceci pour différentes valeurs du paramètre de budget $n \in \{50, 100, 200, 500\}$. Les moyennes empiriques des performances de BOP (en termes de somme de récompenses non actualisées reçues) sont données à la table 1 pour chaque valeur du paramètre n. L'erreur standard est de l'ordre de 2 à 5. On donne aussi dans la même table les performances obtenues par d'autres algorithmes BRL dans des conditions identiques (performances tirées de la littérature).

Tout d'abord, on observe que les performances de BOP s'améliorent avec n. Ensuite, on observe que l'algorithme BOP avec $n = 500$ offre de meilleures performances que les autres algorithmes, à l'exception de BEB (avec une valeur affinée du paramètre β) et BVR, qui font mieux sur ce

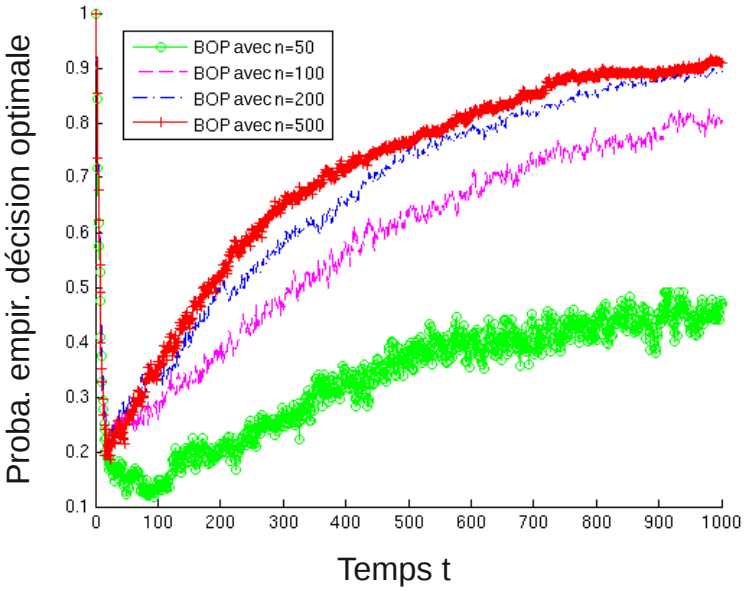

FIGURE 4 – Probabilité empirique de prendre la décision optimale (action $a^{(1)}$) en fonction du temps (l'action $a^{(1)}$ étant optimale pour tous les états).

problème. Notons que l'optimalité Bayésienne est différente de l'optimalité par rapport au MDP sous-jacent, il n'est donc pas surprenant que des algorithmes soient plus effiaces que BOP (qui semble d'ailleurs être proche de l'optimal Bayésien avec $n = 500$). On donne également à la figure 4 l'évolution dans le temps de la probabilité empirique (calculée sur 500 essais) que BOP prenne une décision optimale pour $n \in \{50, 100, 200, 500\}$. A titre informatif, un essais de 1000 pas de temps de BOP prend environ 10 heures (respectivement une heure, 20 minutes et 5 minutes) sur un coeur de processeur d'une machine Linux récente avec $n = 500$ (respectivement $n = 200$, $n = 100$ et $n = 50$) en utilisant Matlab®.

8 Conclusions et perspectives

Cet article décrit l'algorithme BOP (de l'anglais "Bayesian Optimistic Planning"), un nouvel algorithme d'apprentissage par renforcement indirect, qui étend le principe de l'algorithme OP-MDP (Busoniu *et al.* (2011); Busoniu & Munos (2012)) au contexte où le modèle de transition du processus de décision de Markov est initialement inconnu et doit être appris par intéractions.

Dans cet article, on a considéré un espace d'état fini, mais on pourrait étendre BOP au cas où l'espace d'état est infini en contraignant le facteur de branchement des BA-arbres. En termes plus théoriques, il serait intéressant d'analyser quels liens l'exposant de quasi-optimalité du BA-MDP entretient avec l'exposant de quasi-optimalité du MDP sous-jacent.

Remerciements

Raphael Fonteneau est chargé de recherche du F.R.S. - FNRS (Belgique). Lucian Buşoniu est chargé de recherche du CNRS (France). Nous remercions les projets européens (FP7/2007-2013) n°216886 (PASCAL2) and n°270327 (CompLACS), ainsi que le PAI DYSCO (Belgique). Nous remercions également Olivier Nicol pour son aide précieuse.

Références

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

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

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

Auer P., Cesa-Bianchi N. & Fischer P. (2002). Finite time analysis of multiarmed bandit problems. *Machine Learning*, **47**, 235–256.

Bellman R. (1957). *Dynamic Programming*. Princeton University Press.

Brafman R. & Tennenholtz M. (2003). R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, **3**, 213–231.

Bubeck S. & Munos R. (2010). Open loop optimistic planning. In *Conference on Learning Theory (COLT)*, p. 477–489.

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

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

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

Castro P. & Precup D. (2010). Smarter sampling in model-based Bayesian reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, p. 200–214.

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

Coulom R. (2007). Efficient selectivity and backup operators in Monte-Carlo tree search. *Computers and Games*, p. 72–83.

Dearden R., Friedman N. & Russell S. (1998). Bayesian Q-learning. In *National Conference on Artificial Intelligence*, p. 761–768.

Dimitrakakis C. (2008). Tree exploration for Bayesian RL exploration. In *International Conference on Computational Intelligence for Modelling Control & Automation*, p. 1029–1034.

Dimitrakakis C. & Lagoudakis M. G. (2008). Rollout sampling approximate policy iteration. *Machine Learning*, **72**, 157–171.

Duff M. (2002). *Optimal Learning : Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts Amherst.

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

Fonteneau R., Busoniu L. & Munos R. (2013). Optimistic planning for belief-augmented Markov decision processes. In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*.

Gelly S., Wang Y., Munos R. & Teytaud O. (2006). *Modification of UCT with Patterns in Monte-Carlo Go*. Rapport interne, INRIA RR-6062.

Gittins J. (1989). *Multiarmed Bandit Allocation Indices*. Wiley.

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

Hren J. & Munos R. (2008). Optimistic planning of deterministic systems. *Recent Advances in Reinforcement Learning*, p. 151–164.

Ingersoll J. (1987). *Theory of Financial Decision Making*. Rowman and Littlefield Publishers, Inc.

Jaksch T., Ortner R. & Auer P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, **11**, 1563–1600.

Kocsis L. & Szepesvári C. (2006). Bandit based Monte-Carlo planning. *Machine Learning : ECML 2006*, p. 282–293.

Kolter J. & Ng A. (2009). Near-bayesian exploration in polynomial time. In *International Conference on Machine Learning (ICML)*, p. 513–520.

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

Munos R. (2012). *The optimistic principle applied to games, optimization and planning : Towards Foundations of Monte-Carlo Tree Search*. Rapport interne.

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

Ortner R. & Auer P. (2007). Logarithmic online regret bounds for undiscounted reinforcement learning. In *Neural Information Processing Systems (NIPS)*.

Peters J., Vijayakumar S. & Schaal S. (2003). Reinforcement learning for humanoid robotics. In *IEEE-RAS International Conference on Humanoid Robots*, p. 1–20 : Citeseer.

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

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

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

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

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

Sutton R. (1988). Learning to predict by the methods of temporal difference. *Machine Learning*, **3**, 9–44.

Sutton R. & Barto A. (1998). *Reinforcement Learning*. MIT Press.

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

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