Robust and Explorative Behavior in Model-based Bayesian Reinforcement Learning

Toru Hishinuma
Department of Aeronautics and Astronautics
Kyoto University
Kyoto, Japan 615-8540
Email: hishinuma.toru.43n@st.kyoto-u.ac.jp

Kei Senda
Department of Aeronautics and Astronautics
Kyoto University
Kyoto, Japan 615-8540
Email: senda@kuaero.kyoto-u.ac.jp

***Abstract*—This paper considers a tractable simplified problem of model-based Bayesian reinforcement learning (BRL) in terms of real-world samples, computational complexity, and target uncertainties. Robust control and adaptive control are two of the most successful and tractable conventional control design theories against uncertainties in various domain, while they have contrasting ideas. We show that both theories can be explained in a unified manner by approximation model-based BRL algorithms. We propose a forward search tree with robust solutions as a simplified tractable problem, which explicitly includes both theories at the same time. While the structure of the problem has been already seen in a branch and bound method, we provide a novel analysis of the behaviors resulting from it. Through a simple example, we compare the solutions of the proposed problem with the optimal BRL solution and the two conventional approaches, and discuss the interpretation.**

I. Introduction

Control of an unknown system requires two steps: (i) system identification based on sample data, and (ii) control design for the identified model. If the difference between a real system and an identified model is not negligible, then the control design needs to take into account "uncertainties."

Robust control [1] and adaptive control [2] are two of the most successful tractable control design approaches against uncertainties in various domains. Robust control achieves a task using a fixed controller, that allows uncertainties within a certain set. On the other hand, adaptive control reduces real-world uncertainties by online identification, and adapts a controller to the identified model to achieve the task. Therefore, they have different design concepts to deal with uncertainties. In general, it is not clear to determine which approach is better, because the answer depends on what uncertainties must be considered (i.e. the accuracy of system identification).

Reinforcement Learning (RL) is a promising approach to the question, because RL treats both system identification and control design at the same time. RL is a framework within which an agent learns a policy or a controller through trial and error in a real environment [3]. Online sampling process by trial and error corresponds with system identification, said to be "exploration." Finding a policy from collected samples corresponds with control design, called "exploitation." While more samples improve the policy, an agent is forced to pay more exploration cost to obtain them. This is a challenging issue in RL, known as an "exploration-exploitation tradeoff".

However, RL is often intractable in physical domain. As noted in [4], so many samples and calculations are needed due to "the curse of dimensionality" in a large-scale task. Especially in the robotic setting, it is unrealistic to collect too many real-world samples, since they are expensive in terms of time and physical labor (the curse of real-world sample). While introducing models can reduce the number of needed real-world samples, it may lead to narrowing the range of the acceptable uncertainties and often increasing computational complexity. For example, learning only a model needs no real-world samples, but the resulting performance cannot be guaranteed in a real system because of the difference between the model and the real system (the curse of under-modelling and model uncertainties). Therefore, it is important to find a feasible solution in terms of computational complexity, real-world samples, and supposed uncertainties.

In this paper, we consider RL applicable in the physical domain using a model-based Bayesian RL (BRL) framework. The two conventional and tractable approaches for a physical system, robust control and adaptive control, can be explained as approximation model-based BRL algorithms. We propose "an online planning for a forward search tree with leaves given by robust solutions" as a promising approximation BRL approach. This simplified problem explicitly includes the two contrasting approaches against uncertainties, robust and adaptive. Therefore, the idea of the simplified problem has the potential to be applicable to a physical system and to choose a better approach against the uncertainties from the solution of the problem. In this paper, we provide a novel analysis of the behaviors resulting from the simplified problem structure, and discuss the interpretation through a simple example.

In II, we give RL and model-based BRL setting, and refer to related works. In III, we interpret robust and adaptive controllers from the viewpoint of BRL, and describe the proposed approach and the relationship with branch and bound method. In IV, we investigate a simple example to discuss about behaviors resulting from the proposed approach, two conventional approaches, and the optimal BRL solution. Finally, V concludes this paper.

II. Formulation and related works

A. Markov Decision Process and Reinforcement Learning

In this paper, we assume to consider a reinforcement learning (RL) problem whose environment is formulated as a time-independent discrete-time finite Markov decision process (MDP) [5]. A discrete-time finite MDP is defined by $(\mathcal{S},\ \mathcal{U},\ p,\ g)$. The finite state set $\mathcal{S}$ is composed of N states denoted by $s_1, s_2, \cdots, s_N$ and an additional termination state s_0. The finite action set $\mathcal{U}$ is composed of K actions denoted by $u_1, u_2, \cdots, u_K$. If an agent is in state s_i and choose action u_k at time t, it will move to state s_j at time $t+1$ and incur one-step cost $g(s_i, u_k, s_j)$ within transition probability $p_{ij}(u_k) = \Pr(s_j|s_i, u_k)$. We assume that $p_{ij}(u_k)$ is time-independent and dependent only on current state s_i and action u_k explicitly. There is a cost-free termination state s_0, where $p_{00}(u_k) = 1,\ g(s_0, u_k, s_0) = 0,\ \forall u_k$.

Policy π is an action selection rule of an agent. In particular, a stationary policy chooses current action u^t depending only on current state s^t, so that history and time are not taken into account explicitly. A deterministic stationary policy is a mapping from states to actions. A stochastic stationary policy is given by action selection probability $\pi(s_i, u_k) = \Pr(u_k|s_i, \pi)$.

In this study, we deal with a stochastic shortest path problem [5] where the cost accumulates indefinitely without discount and a termination state exists. A standard criterion for evaluating the performance of policy π is its expected cumulative cost, called J-factor for a MDP. It is defined as

$$J^\pi(s_i) \equiv E\left[\sum_{t=t_0}^{\infty} g(s^t, u^t, s^{t+1}) \,\middle|\, s^{t_0} = s_i, \pi\right], \qquad (1)$$

where $E[\,\cdot\,|s_i, \pi]$ denotes the conditional expectation over sequence $[u^{t_0}, s^{t_0+1}, \cdots]$ given policy π and state s_i at initial time t_0.

Stationary policy π is said to be "proper" if it satisfies $J^\pi(s_i) < \infty$ for all s_i [5]. Any stationary policies that are not proper are said to be improper, yielding infinite J-factor at least in one state. In other words, an agent following an improper policy will never reach to the termination state from at least one state. A proper and an improper policies are interpreted as a stable and an unstable controllers, respectively.

An optimal policy for a MDP is a policy π^*, such that $J^{\pi^*}(s_i) = \min_\pi J^\pi(s_i)$ for all states. The optimal J-factor is defined as $J^*(s_i) \equiv J^{\pi^*}(s_i)$. If $p_{ij}(u_k)$ is known, then an optimal policy for the MDP can be obtained only by offline numerical computation, called planning. It is known that a stationary deterministic policy can be optimal for a MDP [6].

However, if $p_{ij}(u_k)$ is unknown, an agent cannot calculate the expectation operator in equation (1) offline. In this case, an agent needs to estimate $p_{ij}(u_k)$ directly interacting with the unknown MDP, and evaluate equation (1) based on the estimation. While higher precision estimation of $p_{ij}(u_k)$ leads to more accurate calculation of equation (1) and better performance, it forces an agent to pay more interaction cost. This dilemma is known as an "exploration-exploitation tradeoff." A standard RL problem considers how to take the tradeoff through finding an optimal solution for a MDP.

B. Bayes-Adaptive MDP and model-based Bayesian RL

Bayes-Adaptive MDP (BAMDP) is defined as an extension of MDP [7]. Let $\theta \in \Theta$ be an unknown time-invariant parameter vector specifying MDP. In this study, we assume that θ changes only transition probability, $p_{ij}(u_k;\theta)$. The parameter set Θ is composed of all possible MDPs. We assume that an agent cannot directly observe unknown θ, while it has model $p_{ij}(u_k;\theta)$ (i.e. a mapping from a parameter to transition probability) for Θ in advance. Belief $b^t(\theta)$ is a conditional distribution over Θ at time t, and updated according to the Bayes rule inside an agent. Specifically, it is prior (at initial time t_0) or posterior (at any other time) distribution, depending on an initial knowledge and online transition history. Hyper-state (s_i, b) is a pair of state s_i and belief b. Then, BAMDP is described by $(\mathcal{S},\ \mathcal{U},\ \Theta,\ p,\ g)$.

If there is no prior knowledge for a system, then the unknown parameter is transition probability itself. That is, an agent does not know $\theta_{ikj} = p_{ij}(u_k)$, and there are N^2K independent parameters. On the other hand, we know that a physical system (e.g. a robot) follows continuous governing equations. If we derive a discrete system by quantization of the original continuous system, then the dynamics also depends on the parameters of the original governing equations. In this case, the MDP can be characterized by much less than N^2K.

In the BAMDP, the performance of policy π is evaluated in a similar way. J-factor for a BAMDP is defined as

$$\bar{J}^\pi(s_i, b^{t_0}) \equiv E\left[\sum_{t=t_0}^{\infty} g(s^t, u^t, s^{t+1}) \,\middle|\, s^{t_0} = s_i, b^{t_0}, \pi\right] \qquad (2)$$

$$= E\left[J^\pi(s_i;\theta) \,\middle|\, b^{t_0}\right] \qquad (3)$$

where $E[\,\cdot,|s_i, b^{t_0}, \pi]$ in equation (2) denotes the conditional expectation over sequence $[u^{t_0}, s^{t_0+1}, \cdots]$ given policy π and hyper-state (s_i, b^{t_0}) at initial time t. In equation (3), $E[\,\cdot\,|b^{t_0}]$ denotes the expectation over MDP parameter θ drawn from belief b^{t_0}, $J^\pi(s_i;\theta)$ denotes J-factor for θ given π and s_i.

An optimal policy for a BAMDP is also defined by $\bar{J}^{\pi^*}(s_i, b^{t_0}) = \min_\pi \bar{J}^\pi(s_i, b^{t_0})$. It is known that an optimal policy for a BAMDP is a deterministic policy mapping from a hyper-state to an action. It is a non-stationary policy since belief b^t depends on online transition sequences up to time t.

The goal of model-based Bayesian RL (BRL) problem is to find a policy π minimizing equation (2). In this setting, an agent knows model $p_{ij}(u_k;\theta)$ and prior distribution b^{t_0} of unknown parameter θ in advance. Therefore, it is possible to optimize equation (2) only by offline computation, in principle. However, since it is computationally intractable to obtain an exact optimal BRL solution, an approximation algorithm is often used to find a feasible solution, as shown in II-D.

C. Challenges in Physical Domain

As noted in [4], RL is generally a difficult problem and many of its challenges are particularly apparent in physical domain such as robotics.

1) Dimensionality: As in other fields, RL for a physical system suffers from "the curse of dimensionality." The state-action space of physical systems is inherently continuous and tends to be high-dimensional. Then, we usually use discretization or function approximation. As the number of dimensions increases, exponentially more data and computation are needed to cover the whole state-action space.

2) Real-World Samples: We assume that an agent (or a robot) needs to know real-world uncertainties, such as friction or measurement precision. Such an uncertainty is unique to each agent facing a real task. The agent must interact with the real environment by itself because experiences from other agents or simulation are inappropriate to identify the real uncertainties. However, trial and error using a real robot takes a high cost in terms of time and physical labor. If a large amount of trial and error is needed, it is difficult to obtain effective solution within a reasonable time frame, because a real robot behavior cannot be sped up unlike simulation. Additionally, a robot should avoid a behavior that might cause a heavy damage to itself. Therefore, there is a limit to real-world interactions, often referred to as "the curse of real-world samples" [4].

3) Model Uncertainty: Using models as simulators can reduce the cost of real-world samples. However, it is difficult to create a sufficiently accurate model of the real task in advance due to unpredictable uncertainties, which is our first motivation to study RL. This is said to be "the curse of under-modelling and model uncertainties" [4]. For this reason, we focus on the model-based BRL setting, in which a model is used as our initial knowledge while uncertainties partially remain. Additionally, introducing models often requires more computational complexity, especially in the BRL setting. Since an agent has to execute an algorithm in real time, some approximation approach is needed.

D. Related Works

There are various classes of approximation algorithms in the model-based BRL setting [8]. Online planning approaches alternate planning and execution, so that planning resources are focused on actually observed histories. Some aspects of them are described in II-D1, II-D2, and II-D3. On the other hand, offline approaches compute the policy a priori, for any possible hyper-state. II-D4 and II-D5 are offline approaches.

1) Sampling from Belief: For example, Bayesian DP [9] periodically samples one MDP parameter θ from belief b^t, solves the online planning problem for the sampled MDP, and chooses actions in the real environment using the resulting policy. If sampled θ is similar to the real system, then the resulting policy is likely to work well. If not, an unsuccessful transition sequence might be observed, and it indicates that sampled θ is not similar to the real system. Belief b^t is updated based on this unsuccessful information, and θ nearer to the real system tends to be re-sampled from updated b^t. Finally, the resulting policy for sampled θ works well in the real system.

2) Guiding Exploration: For example, Bayesian Exploration Bonus (BEB) [10] introduces an additional cost, which explicitly measures an uncertainty using the number of samples of each (s_i, u_k). In the early stage where the additional cost is dominant, an agent is likely to choose explorative behavior in order to improve it than the original evaluation function. As the number of samples increases, the influence of the additional cost decays. Finally, the exploitation using sufficiently collected samples becomes successful in the real environment. In this case, the additional cost serves as an exploration guide to decrease an uncertainty.

A similar idea appears in algorithms based on "optimism", which give a special weight to an uncertainty in order to guide behaviors. For example, BOSS [11] samples some parameters from belief b^t, makes a virtual optimistic MDP by combining sampled MDPs, and uses the resulting policy in the real environment. Therefore, it is a combined technique of ideas of "guiding exploration" and "sampling from belief." In robotic field, optimistic exploration is used in [12], combined with system identification and model predictive control.

Non-Bayesian PAC algorithms such as E^3[13] also have the same idea to give a special weight to an uncertainty.

3) Forward Search: The idea is to solve a finite horizon problem called "a forward search tree" as an online planning problem. It basically contains all hyper-states reachable from current (s^{t_0}, b^{t_0}) within some fixed horizon denoted by d. This method needs a default value function to be used at the leaf node. A full forward search (i.e. without pruning) can only be achieved over a very short decision horizon, since the number of nodes explored grows exponentially with respect to depth d. Branch and bound is a common method for pruning [14].

4) J-factor Approximation: For example, BEETLE [15] directly approximates J-factors of a BAMDP offline. A set of reachable (s_i, b) is sampled by simulation, point-based value iteration [16] is performed at sampled (s_i, b), and a set of approximation functions called α-functions is constructed. However, the computational complexity for constructing α-functions increases exponentially with the planning horizon.

5) Robust Approach: This approach considers to solve a simplified optimization problem, in which any policies depend only on current state s^t at time t (and prior b^{t_0} at initial time t_0), instead of hyper-state (s^t, b^t). In other words, the problem yields only a stationary policy, which cannot depend on any online information.

It is difficult to minimize the evaluation criterion (2) even if a policy is limited to the stationary class. Therefore, a realistic manner is to convert equation (2) to another form and to find a stationary policy minimizing it. Robust DP [1] assumes the uncertainties in different states to be uncorrelated, and finds a stationary policy with best worst-case performance. Specifically, the evaluation function of the simplified problem is given by $\max_{\theta \in \Theta} J^\pi(s_i; \theta)$ instead of equation (2). For another example, risk-sensitive percentile criterion [17] is minimized.

III. Model-Based BRL Approach for Physical Domain

Considering the three challenges in III-C, we are interested in an approximation model-based BRL algorithm with tractable real-world samples, computational complexity, and supposed uncertainties. In this study, we focus on structures of simplified optimization problems and the behaviors resulting from them. Although approximate representations such as neural networks or solvers such as Monte Carlo method are not discussed here, our result can be combined with them.

In III-A, we make a comparison between behaviors coming from robust and explorative (corresponding to adaptive) strategies. In III-B, we propose a simplified problem, which is a promising approximation approach including both robust and explorative strategies. Finally, III-C refers to the relationship between proposed approach and a branch and bound method.

A. Robust vs Explorative Approaches

From the viewpoint of model-based BRL, we discuss both robust control and adaptive control, that are major conventional approaches against uncertainties (III-A1). Then, strong and weak situations for each approach are described in III-A2.

1) Interpretation as Approximation BRL: Robust control designs a fixed controller that is hard to deteriorate by supposed uncertainties. A robust controller accepts all possible uncertainties at the same time, since its performance has to be guaranteed without using online information. A BRL solution limited to the stationary class in II-D5 is a robust strategy.

On the other hand, adaptive control strategy parametrizes uncertainties, estimates the unknown parameter using online data, and changes the controller based on the estimated parameter. In a standard adaptive control, at first system identification is executed, and then the controller becomes stable. Therefore, we consider that it corresponds with a BRL approach that works successfully after uncertainties are reduced. In this study, we interpret all algorithms as explorative strategies, where the algorithms are ranging from BEB guiding exploration explicitly to Bayesian DP, which implicitly drives exploration by sampling from the posteriori distribution.

Both robust and adaptive controls can be viewed as the approximation approaches to model-based BRL. Therefore, J-factor (2) can measure their performances in a unified manner.

2) Strong and Weak Situations: If one-step cost of a transition to identify the real uncertainties is very large compared to robust performance, the robust strategy tends to be better than the explorative one. In addition, if controller variation may significantly spoil control performance, the explorative strategy is likely to incur large cost due to a controller resulting from the inaccurate estimation with less samples in an early stage, while it rarely happens in the robust strategy.

On the other hand, the explorative strategy is a superior approach where the robust one does not work well. Namely, that is where only small cost is needed to know uncertainties and varying controller can significantly improve the performance.

B. Proposed Approach

There are two major conventional choices to deal with uncertainties: robust (accepting uncertainties) and explorative (reducing uncertainties) control theories. We propose a simplified model-based BRL problem, "a forward search tree with leaves given by robust policies," as an approximation approach which includes the two choices explicitly at the same time.

In this problem, a node in a tree is specified by a pair of hyper-state (s_i, b) and depth d, denoted by (s_i, b, d). In the case of $d = 0$, J-factor of a leaf node is given as

$$\bar{J}'^{*}(s_i, b, 0) = \min_{\pi \in \Pi} E\left[J^{\pi}(s_i; \theta) \;\middle|\; b \right], \tag{4}$$

where $E[\cdot|b]$ denotes the expectation over MDP parameter θ drawn from belief b, and Π is a set of stationary policies. As mentioned in II-D5, the minimization of (4) under the constraints of stationary policies is relatively tractable. Π includes a robust feasible solution of $E[J^{\pi}(s_i; \theta)|b]$ and/or an optimal policy for a certain MDP θ. The minimization operator is likely to choose a robust policy in the face of large uncertainties, or an optimal policy in the face of negligible uncertainties. In general, a stationary policy minimizing equation (4) is different depending on hyper-state (s_i, b).

In the case of $d \geq 1$, J-factor of node (s_i, b, d) satisfies

$$\bar{J}'^{*}(s_i, b, d) = \min_{u_k} E\left[g(s_i, u_k, s_j) + \bar{J}'^{*}(s_j, b', d-1) \;\middle|\; s_i, b, u_k \right],$$

where $E[\cdot|s_i, b, u_k]$ denotes the conditional expectation over next s_j and MDP parameter θ given current $si,\ b$ and u_k. Next belief b' depends on current belief b and current transition (s_i, u_k, s_j), and is calculated by the Bayes rule:

$$b'(\theta) \equiv \Pr(\theta|b, s_i, u_k, s_j) = \frac{p_{ij}(u_k; \theta) b(\theta)}{E[p_{ij}(u_k; \theta')|b]},$$

where $E[\cdot|b]$ denotes the expectation over MDP parameter θ' drawn from belief b. In model-based BRL setting, transition probability model $p_{ij}(u_k; \theta)$ is given as initial knowledge.

This approach is detailed in Algorithm 1. This is a basic full forward search in which equation (4) is used at the leaf nodes. For simplicity, we use $\bar{p}_{ij}(u_k; b) \equiv E[p_{ij}(u_k; \theta')|b]$.

The simplified problem can give explorative behaviors within finite depth d. Furthermore, a set of feasible solutions of the problem includes a behavior following only a stationary policy from current hyper-state. The optimal solution of the problem must be better than or equal to a robust policy in Π. Therefore, the solution of the simplified problem explicitly contains both robust and explorative strategies.

In a large-scale task such as robotics, even a robust stationary policy for a BAMDP requires a large amount of computation. Therefore, it is a realistic manner to execute offline calculation to obtain only several robust stationary policies to construct Π, and to maintain Π to use online in equation (4). Even when we cannot choose enough depth d due to computational limitation for a large-scale problem, at

Algorithm 1 Forward Search with Robust Leaf

function RobustLeaf-FS (s_i, b, d)
if $(d = 0)$ **then**
  return $\min_{\pi \in \Pi} E\left[J^{\pi}(s_i; \theta) \,\middle|\, b \right]$
end if
$minQ \leftarrow \infty$
for $u_k \in \mathcal{U}$ **do**
  $q \leftarrow 0$
  for $s_j \in \mathcal{S}$ **do**
    $b'(\theta) \leftarrow \Pr(\theta | s_i, u_k, s_j, b)$
    $q \leftarrow q + \bar{p}_{ij}(u_k; b) \times g(s_i, u_k, s_j)$
    $q \leftarrow q + \bar{p}_{ij}(u_k; b) \times$ RobustLeaf-FS$(s_j, b', d-1)$
  end for
  if $(q < minQ)$ **then**
    $minQ \leftarrow q$
  end if
end for
return $minQ$

worst prepared robust performance in Π is guaranteed. This is convenient for RL for a real-world physical system.

C. Relationship with Branch and Bound

It is the idea seen in branch and bound method [14] that uses J-factors of a policy as leaf node values disregarding online information. In the context of POMDP planning, the main purpose is roughly approximating lower bound of an optimal J-factor to reduce the computational complexity, and known as a common pruning technique for a tree. As discussed in III-B, the structure (i.e. to obtain lower bound) of the branch and bound method can be also viewed as dealing with two contrasting approaches against uncertainties at the same time. In this paper, we study the behaviors resulting from the structure through a simple example in the next section.

IV. Example

In this section, we provide an experimental analysis of the behaviors resulting from optimal BRL solution, robust strategy, explorative strategy and proposed approach.

A. Definition

At first, we define a BAMDP. The state space is composed of $s_1,\ s_2,\ s_0$, as illustrated in Fig. 1. The action space is composed of $u_1,\ u_2,\ u_3$. We assume that there are two MDPs specified by time-independent parameters θ_1 and θ_2. The agent does not know which is the real MDP parameter.

The transition probability of each MDP is illustrated in Fig. 2. Let u_1 be an action toward the termination state. If the real MDP parameter is θ_1, then the next state for current (s_1, u_1) is s_0 within probability q_1 $(0 < q_1 \leq 1)$, while that for current (s_2, u_1) fails to be s_0. If the real MDP parameter is θ_2, vice versa. Let u_2 be an action toward the lateral state.

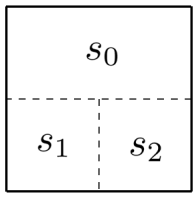

Fig. 1. State space.

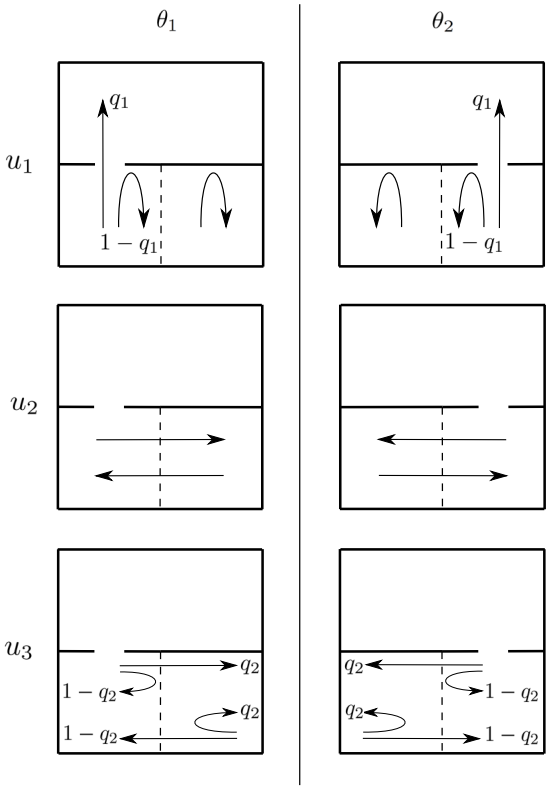

Fig. 2. Transition probability.

In both MDPs, the action deterministically succeeds the move. When an agent chooses u_3, it moves to the state further from s_0 with probability q_2 or the state nearer to s_0 with $1 - q_2$, in each MDP. Then, u_3 can be viewed as an explorative action.

The one step cost depends only on u_k, $g(s_i, u_k, s_j) = g_k$.

The constants characterizing this BAMDP are $q = [q_1,\ q_2]$ and $g = [g_1,\ g_2,\ g_3]$. This BAMDP is a modification of the Tiger problem [18], which is a popular POMDP example, in order to discuss property for MDPs (i.e. reachability to s_0) within BRL framework.

B. Result: Optimal BRL Solution

Using a grid-based approximation scheme, we numerically obtain an optimal BRL solution, which chooses an action depending on hyper-state (s_i, b). For simplicity, a component of a belief at time t is denoted by $b_n^t \equiv b^t(\theta_n)$. There are two MDPs, and a belief is represented as $b = [b_1,\ 1 - b_1]$. We use b_1 instead of b because they have one-to-one correspondence. Therefore, a hyper-state is denoted by (s_i, b_1).

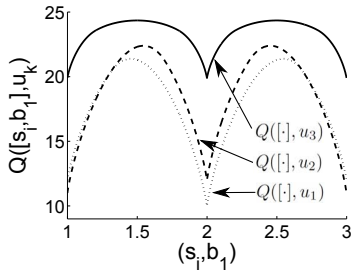

Fig. 3. Optimal Q-factor (g_3 is large).

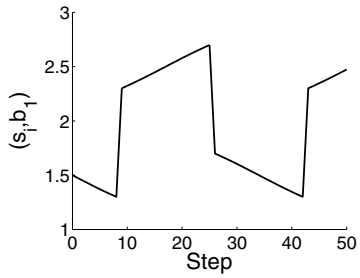

Fig. 4. Optimal behavior (g_3 is large).

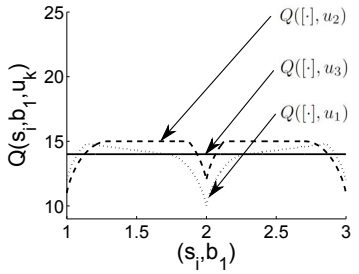

Fig. 5. Optimal Q-factor ($q_2 = 1.0$ and g_3 is small).

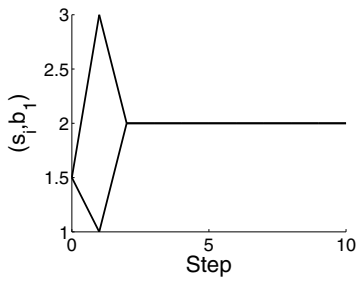

Fig. 6. Optimal behavior ($q_2 = 1.0$ and g_3 is small).

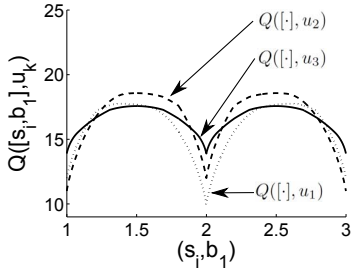

Fig. 7. Optimal Q-factor ($q_2 < 1.0$ and g_3 is small).

In the following figures, we omit termination hyper-state (s_0, b_1), and represent hyper-state (s_i, b_1) as one-dimensional variable $(i+b_1)$, for simplification. Symmetry of the BAMDP leads to the same optimal Q-factor and the same optimal action at hyper-state (s_1, b_1) and $(s_2, 1-b_1)$. In the case of $b_1 = 0.0$ or 1.0, a planning problem for a BAMDP is reduced to one for a MDP, because an agent is absolutely certain which is the real MDP parameter. Therefore, hyper-states $(s_1, 1.0)$ and $(s_2, 0.0)$ have the same optimal action, which repeats only u_1 toward the termination state. There is no problem to equate hyper-state $(s_1, 1.0)$ with $(s_2, 0.0)$ by the one-dimensional reduction.

We examine optimal BRL behaviors for some BAMDP constants (q, g) and show them as follows.

1) When g_3 is large: We show a (numerical) optimal BRL solution when $q = [0.1,\ 0.9]$ and $g = [1,\ 1,\ 9]$. Fig. 3 shows the optimal Q-factor. Fig. 4 illustrates all possible behaviors resulting from the optimal BRL policy from initial hyper-state $(s_1, 0.5)$. As Fig. 4 shows, the optimal solution repeats periodic behavior, in which the agent chooses u_1 8 times from $b_1 = 0.5$ in one state, uses u_2 to move to the other state, chooses u_1 8 times again in the other state, and come back to $b_1 = 0.5$. Therefore, it is a 17 step periodic behavior, which changes according to the BAMDP constant.

2) When $q_2 = 1.0$ and g_3 is small: We show an optimal solution when $q = [0.3,\ 1.0]$ and $g = [1,\ 1,\ 3]$. Fig. 5 shows the optimal Q-factor. Fig. 6 illustrates all possible behaviors resulting from the optimal BRL policy from initial hyper-state $(s_1, 0.5)$. At $t = 0$, the agent chooses u_3 and moves to farther state from s_0. If the next state is s_2, the real MDP parameter must be θ_1 and the agent decides the optimal policy for θ_1. If the next state is s_1, the agent decides the optimal policy for θ_2. At $t = 1$, it selects u_2 to move to s_1. After $t = 2$, it repeats u_1 and attempts to move to s_0.

3) When $q_2 < 1.0$ and g_3 is small: We show a optimal solution when $q = [0.1,\ 0.9]$ and $g = [1,\ 1,\ 3]$. Fig. 7 shows the optimal Q-factor. Fig. 8 illustrates all possible behaviors of the optimal BRL policy from initial hyper-state $(s_1, 0.5)$. At $t = 0$, the agent chooses u_3. If the next state is s_2, then the real MDP parameter is more likely to be θ_1 because $q_2 > 0.5$. If the next state is s_1, then θ_2 is estimated. In each case, the agent selects actions based on estimation result for a while (e.g. it attempts to reach to s_0 via s_1 when θ_1 is estimated). As an agent fails to reach to s_0 after a while, the posterior probability of the estimated MDP decreases. If current one-dimensional variable $(i+b_1)$ exceeds the intersection in Fig. 7, then the agent chooses u_3 again to estimate the real MDP.

C. Result: Proposed Approach

We illustrate behaviors resulting from the simplified problem in III-C. Since this BAMDP is a simple example, we can numerically minimizing J-factor (2) in the class of stationary policy. We use the solution as the leaf value (4).

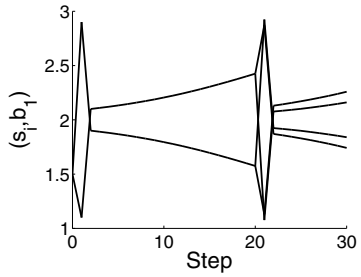

Fig. 8. Optimal behavior ($q_2 < 1.0$ and g_3 is small).

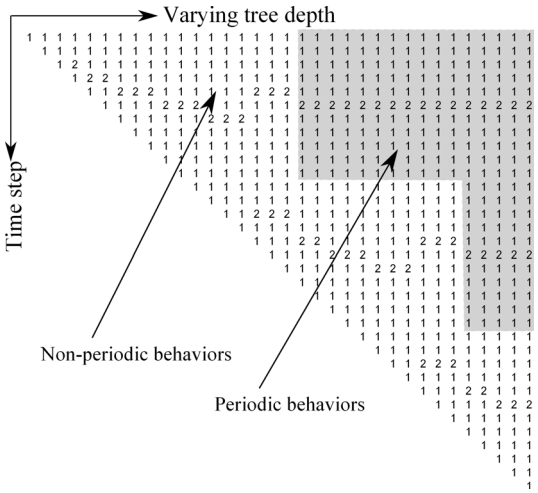

Fig. 9. Behaviors of our approaches (g_3 is very large).

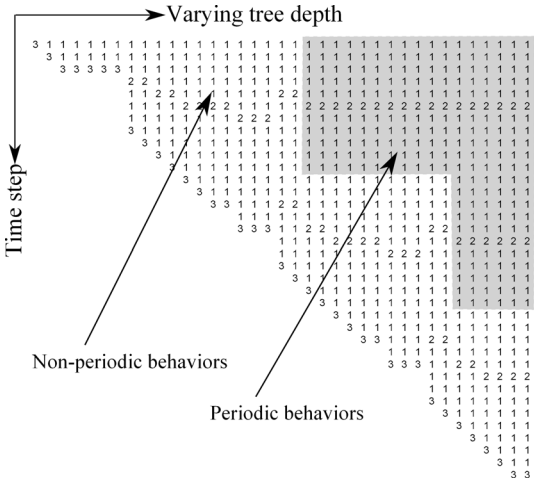

Fig. 10. Behaviors of our approaches (g_3 is large).

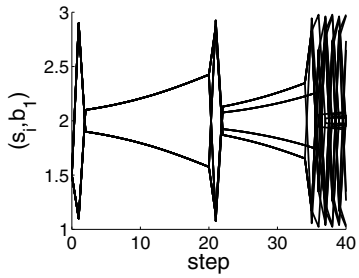

Fig. 11. Behavior of our approach ($q_2 < 1.0$ and g_3 is small).

1) When g_3 is large: At first, we illustrate behaviors resulting from our proposed problem (III-B) when $q = [0.2,\ 1.0]$ and $g = [1,\ 1,\ 10]$. In this case, g_3 is very large and an optimal BRL solution leads to a periodic behavior, alike IV-B1. Fig. 9 shows behaviors when we vary search depth D. This figure describes sequences of the index of action u_k, the horizontal direction represents search depth D, and the vertical one represents time step t of each sequence. Because there is only one next state except for s_0 due to $q_2 = 1.0$, the behaviors are uniquely described according to only sequences $\{u^t\}$.

Next, we illustrate behaviors when $q = [0.2,\ 1.0]$ and $g = [1,\ 1,\ 6]$. An optimal BRL solution also results in a robust behavior in this case. Fig. 10 shows behaviors when we vary search depth D. We omit parts of behaviors after u_3 is selected, because they follow the optimal policy for the confirmed MDP.

Comparing Figs. 4, 9, and 10, we observe: (i) periodic behaviors similar to the optimal BRL solution in the shallow part of the tree, (ii) changing the timing to choose u_2 in the deep part, and (iii) occasional u_3 in the deep parts if explorative cost g_3 is not too large.

2) When $q_2 = 1.0$ and g_3 is small: We show behaviors when $q = [0.1,\ 1.0]$ and $g = [1,\ 1,\ 3]$, equivalent to IV-B2. A forward search tree with finite depth $D \geq 1$ includes a behavior in which u_3 is selected at initial time t_0. After u_3, an agent should follow the (stationary) optimal policy for the confirmed MDP. This behavior is equivalent to the exact optimal BRL solution, and of course the proposed approach cannot have solutions better than it.

3) When $q_2 < 1.0$ and g_3 is small: We show behaviors when $q = [0.1,\ 0.9]$ and $g = [1,\ 1,\ 3]$, equivalent to IV-B3. All possible behaviors for a forward search tree with depth $d = 40$ are shown in Fig. 11.

Comparing Figs. 8 and 11, we observe: (i) explorative behaviors similar to the optimal BRL solution in the shallow part of the tree, and (ii) changing the timing to choose u_3 in the deep part.

D. Conventional Approaches

1) Behaviors from Robust Approach: We describe policies obtained by the robust approach for this BAMDP. For simplicity, we write action selection probability as $\pi_{ik} = \pi(s_i, u_k)$.

We consider property for the two MDPs. In MDP θ_1, the probability from s_2 to s_0 following π within 2 steps is $q_1\pi_{11}[\pi_{22} + (1-q_2)\pi_{23}]$. In MDP θ_2, the probability from s_1 to s_0 following π within 2 steps is $q_1\pi_{21}[\pi_{12} + (1-q_2)\pi_{13}]$. If stationary policy π is proper for θ_1 and θ_2, it satisfies

$$0 < \pi_{i1} < 1, \quad 0 < [\pi_{i2} + (1-q_2)\pi_{i3}], \quad i = 1, 2,$$

because $\pi_{i1} + \pi_{i2} + \pi_{i3} = 1$ $(i = 1, 2)$. Therefore, a proper policy for the BAMDP must be stochastic. If $g_3 \gg g_2$, then a stationary policy satisfies $\pi_{i1} > 0,\ \pi_{i2} > 0,\ \pi_{i3} = 0$.

2) Behaviors from Explorative Approach: The basic idea of explorative strategy is that an agent behaves to decrease uncertainties in an early stage, and it chooses an effective policy for the most probable MDP after the uncertainties become small.

In this BAMDP, b_1 near to 0.5 means a large uncertainty and b_1 near to 0 or 1 means a small uncertainty. For example, if we use information entropy

$$H(b_1) = -b_1 \log b_1 - (1 - b_1) \log(1 - b_1)$$

as a measure of the uncertainty, then an agent is better to choose an action to decrease it. In the case of $q_2 = 1$, u_1 changes H when it cannot reach to s_0 (e.g. decreases at $b_1 = 0.5$), u_2 does not vary H, and u_3 always leads to $H = 0$. In the case of $q = [0.1,\ 0.9]$, u_3 changes H more significantly than u_1 does. H can be used as an additional cost.

E. Discussion

1) Understanding of Periodic Behavior without u_3: The optimal BRL policy repeats u_1 and u_2 periodically when g_3 is large (IV-B1). On the other hand, a proper policy for the two MDPs satisfies $\pi_{i1} > 0,\ \pi_{i2} > 0$ (IV-D1), and chooses u_1 or u_2 at a fixed frequency. These two behaviors are not the same, but has a similar aspect. In the optimal policy, belief b_1 remembers the failure count of u_1 and decides when to choose u_2. In a proper stochastic stationary policy, a frequency indirectly reflects the failure count of u_1. Therefore, we expect that a stochastic stationary policy from robust strategy work well when g_3 is large. We interpret a behavior choosing u_1 or u_2 in a certain proportion as a robust strategy.

2) Understanding of Behavior using u_3: When $q_2 = 1.0$, action u_3 pays explorative cost g_3 and recedes from s_0 to confirm the real MDP. Therefore, u_3 is an action only to decrease the uncertainty. When $q_2 < 1.0$, it is the same situation. The optimal BRL solution in IV-B2 chooses u_3 to confirm the real MDP, and selects actions that are optimal only for the MDP. The optimal BRL solution in IV-B3, also chooses u_3 and changes its policy according to the transition result. Furthermore, an explorative approach in IV-D2 chooses u_3 in an early stage to decrease information entropy H. In this study, we interpret a behavior choosing u_3 when the uncertainty is large in a certain proportion as an explorative strategy.

3) Proposed Approach for Robust vs Explorative: When g_3 is large, the optimal BRL solution is a robust behavior (IV-E1). When g_3 is small, the optimal BRL solution is an explorative behavior (IV-E2). A robust or an explorative behavior is obtained corresponding to g_3 in a unified manner (IV-B). Furthermore, the solution resulting from our simplified problem has the following property: (i) it can chooses better strategy among robust and explorative strategies, and (ii) it is likely to select a more explorative behavior than the optimal BRL solution in the deep part of the tree.

V. Conclusion

We have presented "the forward search tree with leaves given by robust stationary policies", the simplified model-based BRL problem with tractable real-world samples, computational complexity, and supposed uncertainties. This problem has explicitly included two contrasting conventional approaches for uncertainties, robust and adaptive controls. We have discussed behaviors resulting from the structure of the problem through the simple problem. We have provided the new interpretation of robust and explorative behavior in the context of model-based BRL in a unified manner. We show that the solution of the simplified problem can choose a better approach against the uncertainties.

In this paper, we try not to use approximation functions in order to discuss the property resulting from the simplified problem structure. However, we need to depend on additional approximation functions for a real physical system. The future work of this paper is to apply this simplified problem together with some additional approximation functions to verify the practicality.

Acknowledgment

A part of this work has been financially supported by a grant in aid for Scientific Research from the Ministry of Education, Science, Culture, and Sports of Japan.

References

[1] A. Nilim and L. El Ghaoui: Robust Control of Markov Decision Processes with Uncertain Transition Matrices; *Operations Research*, Vol. 53, No. 5, pp. 780–798 (2005).

[2] H. Kaufman et al.: *Direct Adaptive Control Algorithms: Theory and Application (2nd ed)*, Springer-Verlag (1998).

[3] R. S. Sutton and A. G. Barto: *Reinforcement Learning: An Introduction*, MIT Press (1998).

[4] J. Kober et al.: Reinforcement Learning in Robotics: A Survey; *Int. J. Robotics Research*, Vol. 32, No. 11, pp. 1238–1274 (2013).

[5] D. P. Bertsekas and J. N. Tsitsiklis: *Neuro-Dynamic Programming*, Athena Scientific, (1996).

[6] M. L. Puterman: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, A Wiley-Interscience publication (1994).

[7] M.O.G. Duff: *Optimal Learning: Computational Procedures For Bayes-Adaptive Markov Decision Processes*, PhD thesis, University of Massachusetts Amherst (2002).

[8] M. Ghavamzadeh et al.: *Bayesian Reinforcement Learning: A Survey*, Now Publishers Inc. (2015).

[9] M. Strens: A Bayesian framework for reinforcement learning; *Proc. ICML*, pp. 943–950 (2000).

[10] J. Z. Kolter and A. Y. Ng: Near-Bayesian exploration in polynomial time; *Proc. ICML*, pp. 513–520 (2009).

[11] J. Asmuth et al.: A Bayesian Sampling Approach to Exploration in Reinforcement Learning; *Proc. UAI*, pp. 19–26 (2009).

[12] C. Xie et al.: Model-based Reinforcement Learning with Parametrized Physical Models and Optimism-Driven Exploration; arXiv preprint arXiv:1509.06824 (2015).

[13] M. Kearns and S. Singh: Near-optimal reinforcement learning in polynomial time; *Proc. ICML*, pp. 260–268 (1998).

[14] S. Ross et al.: Online Planning Algorithms for POMDPs; *J. Artificial Intelligence Research*, Vol. 32, pp. 663–704 (2008).

[15] P. Poupart et al.: An analytic solution to discrete Bayesian reinforcement learning; *Proc. ICML*, pp. 697–704 (2006).

[16] J. Pineau et al.: Point-based value iteration: An anytime algorithm for POMDPs; In *IJCAI*, Vol. 3, pp. 1025–1032 (2003).

[17] E. Delage and S. Mannor: Percentile optimization for Markov decision processes with parameter uncertainty; *Operations Research*, Vol. 58, No. 1, pp. 203–213 (2010).

[18] L. P. Kaelbling et al.: Planning and acting in partially observable stochastic domains; *Artificial intelligence*, Vol. 101, No. 1, pp. 99-134 (1998).