2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Improving Offline Value-Function Approximations for POMDPs by Reducing Discount Factors

Yi-Chun Chen,[1] Mykel J. Kochenderfer,[2] and Matthijs T. J. Spaan[3]

***Abstract*— A common solution criterion for partially observable Markov decision processes (POMDPs) is to maximize the expected sum of exponentially discounted rewards, for which a variety of approximate methods have been proposed. Those that plan in the belief space typically provide tighter performance guarantees, but those that plan over the state space (e.g., QMDP and FIB) often require much less memory and computation. This paper presents an encouraging result that shows that reducing the discount factor while planning in the state space can actually improve performance significantly when evaluated on the original problem. This phenomenon is confirmed by both a theoretical analysis as well as a series of empirical studies on benchmark problems. As predicted by the theory and confirmed empirically, the phenomenon is most prominent when the observation model is noisy or rewards are sparse.**

I. INTRODUCTION

The problem of sensing and acting in an uncertain environment for robots can be formalized as a partially observable Markov decision process (POMDP) [1], [2]. Because solving POMDPs exactly is computationally intractable in general, there has been interest in approximation methods. One class of methods involves approximating the value function, which is known to be piecewise linear and convex and can be represented as a set of so-called alpha vectors [3]. SARSOP is a state-of-the-art offline method for computing a set of alpha vectors that approximates the optimal value function [4]. It is an iterative algorithm that plans in the space of belief states. Other algorithms, such as QMDP (Q-function approximation) [5], UMDP (Unobservable MDP) [3], and FIB (Fast Informed Bound) [3], do not plan in the space of beliefs, but plan over the state space. Consequently, these algorithms require much less memory and computation than belief-space planners, and have been widely used in robot control and manipulation [6], [7], [8], even if they are not able to represent optimal policies.

This paper shows that lowering the discount factor while computing alpha vectors can, surprisingly, improve the performance of state-space planners. The state-space planners, QMDP, UMDP, and FIB, coarsely approximate POMDP value functions by associating each action with exactly one alpha vector. This inaccurate approximation can introduce error in accounting for future events in value iteration. Longer effective horizons can lead to higher error due to the inaccuracy of the model. As a result, it can be appropriate to shorten the effective horizon, i.e., to reduce the planning discount factor, as a trade-off between rewards from future events and errors in the model. The benefits of decreasing the discount factor has been observed in other contexts, including approximate dynamic programming for MDPs [9], reinforcement learning [10], and shallow planning in fully observable settings [11], but the phenomenon has not yet been studied in a POMDP context.

Our main contribution is showing that in the context of partial observability with a known model, planning with a lower discount factor can significantly improve the quality of the resulting policy. A theoretical analysis shows that the error between the true value function and the approximate one can be bounded and further reduced when the latter is planned with a lowered discount factor. In addition, when rewards are sparse, a tighter error bound is obtained.

Our experiments involve seven benchmark problems with three state-space planners, QMDP, UMDP, and FIB. A striking result is that in several problem domains, the policies generated with a lower discount factor with these methods nearly match SARSOP's performance at only a fraction of the computational expense. In addition, when observation noise is high or rewards are sparse, the phenomenon is more significant, as predicted by our theoretical analysis and demonstrated experimentally.

II. VALUE-FUNCTION APPROXIMATIONS

Associated with a POMDP is a set of states S, a set of actions A, and set of observations Ω. The immediate reward for taking action a in state s is $R_{s,a}$. The probability of transitioning to state s' from state s after taking action a is $T_{s,a}^{s'}$. The probability of observing o after taking action a and ending up in state s' is $O_{s',a}^{o}$. The objective is to find a policy for selecting actions given past observation histories that maximizes the expected sum of exponentially discounted rewards, where the discount factor is denoted by γ.

QMDP [5] is an offline value-approximation method and also a state-space planner. It creates a set of alpha vectors, one for each action, based on the state-action value function $Q(s,a)$. In addition, alpha vectors are computed as follows:

$$\alpha_a^{(k+1)}(s) = R_{s,a} + \gamma \sum_{s'} T_{s,a}^{s'} \max_{a'} \alpha_{a'}^{(k)}(s'). \qquad (1)$$

With converged alpha vectors $\alpha_a(s)$, the policy for a belief state b is $a^* = \text{argmax}_a \sum_s \alpha_a(s) b(s)$. QMDP assumes the

[1]Yi-Chun Chen is a PhD student at UCLA Anderson School of Management, USA. yi-chun.chen.phd@anderson.ucla.edu

[2]Mykel J. Kochenderfer is an Assistant Professor of Aeronautics and Astronautics and, by courtesy, of Computer Science at Stanford University, USA. mykel@stanford.edu

[3]Matthijs T. J. Spaan is an Associate Professor at Delft University of Technology, Delft, The Netherlands. m.t.j.spaan@tudelft.nl

978-1-5386-8094-0/18/$31.00 ©2018 IEEE

full observability of the next state, and therefore the right-hand side of Eq. (1) is simply the MDP update rule. QMDP has been widely used in robot manipulation [6], [7], [8].

UMDP [3] assumes no observability of future states. The alpha vectors for each action are obtained by:

$$\alpha_a^{(k+1)}(s) = R_{s,a} + \gamma \text{max}_{a'} \sum_{s'} T_{s,a}^{s'} \alpha_{a'}^{(k)}(s'). \quad (2)$$

Due to the assumption of no observability, the maximization operator on the right-hand side of Eq. (2) is executed before the summation over s'.

Finally, FIB [3] takes the partial observability of next states into account by computing the following alpha vectors:

$$\alpha_a^{(k+1)}(s) = R_{s,a} + \gamma \sum_o \max_{a'} \sum_{s'} O_{s',a}^o T_{s,a}^{s'} \alpha_{a'}^{(k)}(s'). \quad (3)$$

Update operators can defined for QMDP, UMDP, and FIB. For example, QMDP approximates H as follows:

$$\begin{aligned} \hat{V}^{(i+1)}(b) = \max_a \sum_s b(s)[R_{s,a} + \\ \gamma \sum_{s'} T_{s,a}^{s'} \cdot \max_{a'} \alpha_{a'}(s')] \equiv H_{\text{QMDP}} \hat{V}^{(i)}(b). \end{aligned} \quad (4)$$

There are other offline approximation approaches, such as point-based value iteration [12], Perseus [13], and SARSOP [4]. In contrast to QMDP, UMDP, and FIB, which are state-space planners, these point-based methods plan in the belief space. Alpha vectors are computed at several belief points, and the number of belief points typically increases during the planning process, usually resulting in many more alpha vectors than one for each action. For example, in the Maze20 problem [3], SARSOP produces thousands of alpha vectors, while QMDP uses only six alpha vectors. Belief-space planners can provide much better policies for POMDP problems, but often at significant computational cost. Another advantage of state-space planners is that they do not require knowledge of the initial belief, while point-based methods require it for planning.

III. Theoretical Analysis

This section presents a theoretical analysis that shows that when there exists approximation error introduced by value-function approximation methods, the error can be reduced by planning with a lower discount factor.

A. Error Bound of Approximate Value Functions

Without loss of generality, assume all rewards are non-negative. We denote the true discount factor γ and the lowered discount factor γ' with $\gamma \geq \gamma'$. Let V_γ be the optimal value function with the true discount factor γ, and $U_{\gamma'}$ be the approximate value function planned using γ' by either QMDP, UMDP, or FIB as in Eq. (4).

The difference between the true value function and the approximate value function planned using a lowered discount factor is $\|V_\gamma - U_{\gamma'}\|_\infty$, where $\|\cdot\|_\infty$ is the infinity norm. This error can be bounded as follows:

$$\|V_\gamma - U_{\gamma'}\|_\infty \leq \|V_\gamma - V_{\gamma'}\|_\infty + \|V_{\gamma'} - U_{\gamma'}\|_\infty \quad (5)$$

where $e_d \equiv \|V_\gamma - V_{\gamma'}\|_\infty$ is the discount error that measures the difference between optimal value functions of true discount factor and lowered discount factor, and $e_a \equiv \|V_{\gamma'} - U_{\gamma'}\|_\infty$ is the approximation error due to approximation methods. Equation (5) follows the triangle inequality in infinity-norm metric space.

According to Theorem 2 of [9], the discount error e_d can bounded as follows:

$$e_d = \|V_\gamma - V_{\gamma'}\|_\infty \leq (\gamma - \gamma') \cdot V_{\text{max}}(\gamma)/(1-\gamma'), \quad (6)$$

where $V_{\text{max}}(\gamma)$ is the largest possible value of the value function with discount factor γ. Normally, this is set to be $r_{\text{max}} + r_{\text{max}}\gamma + r_{\text{max}}\gamma^2 + \ldots = r_{\text{max}}/(1-\gamma)$, where r_{max} is the largest immediate reward. This value will be evaluated differently later in the sparse reward scenario.

The approximation error $e_a(\gamma') \equiv \|V_{\gamma'} - U_{\gamma'}\|_\infty$ can also be bounded. Let $e_a^{(k)}(\gamma')$ be the approximation error after k applications of the Bellman update, $V_{\gamma'}^{(k)}$ and $U_{\gamma'}^{(k)}$ be the optimal and approximate value functions after k times of value iteration, respectively, and H and H^{ap} be the exact and approximate update operators. Therefore,

$$\begin{aligned} e_a^{(k)} &= \|V_{\gamma'}^{(k)} - U_{\gamma'}^{(k)}\|_\infty = \|HV_{\gamma'}^{(k-1)} - H^{ap}U_{\gamma'}^{(k-1)}\|_\infty \\ &\leq \|HV_{\gamma'}^{(k-1)} - HU_{\gamma'}^{(k-1)}\|_\infty \\ &\quad + \|HU_{\gamma'}^{(k-1)} - H^{ap}U_{\gamma'}^{(k-1)}\|_\infty \\ &\leq \gamma' \cdot e_a^{(k-1)} + \|HU_{\gamma'}^{(k-1)} - H^{ap}U_{\gamma'}^{(k-1)}\|_\infty. \end{aligned}$$

The first and second inequalities come from the triangle inequality in the metric space and contraction property of Bellman operator, respectively. We define $\epsilon_0 = \max\{\|HU - H^{ap}U\|_\infty \mid U = \max_{a \in A} b \cdot \alpha_a\}$, the difference between exact operator H and approximate operator H^{ap} ranging over all possible value functions that are composed of $|A|$ alpha vectors. Given any $|A|$ alpha vectors composing U, HU and $H^{ap}U$ can be computed through standard methods, such as Eq. (4). Thus, ϵ_0 is just the upper bound of $\|HU - H^{ap}U\|_\infty$, and we have

$$\begin{aligned} e_a^{(k)} &\leq \gamma' \cdot e_a^{(k-1)} + \epsilon_0 \leq \gamma' \cdot (\gamma' \cdot e_a^{(k-2)} + \epsilon_0) + \epsilon_0 \\ &\leq \gamma'^k e_a^{(0)} + \epsilon_0 \cdot (1 - \gamma'^k)/(1-\gamma'). \end{aligned}$$

Furthermore, since each component of α_a is less or equal to $V_{\text{max}}(\gamma')$, we can normalize ϵ_0 by defining $\epsilon = \epsilon_0 / V_{\text{max}}(\gamma')$ and have $\epsilon \leq 1$. The value of ϵ depends on the problem and approximation method. The smaller the ϵ, the closer the approximation method is to the exact method. As a result,

$$\begin{aligned} e_a^{(k)} &\leq \gamma'^k e_a^{(0)} + \epsilon_0 \cdot (1 - \gamma'^k)/(1-\gamma') \\ &= \gamma'^k e_a^{(0)} + \epsilon \cdot V_{\text{max}} \cdot (1 - \gamma'^k)/(1-\gamma'). \end{aligned}$$

By taking the limit of k, $e_a(\gamma')$ is bounded by

$$e_a(\gamma') = \|V_{\gamma'} - U_{\gamma'}\|_\infty \leq \frac{\epsilon}{1-\gamma'} \cdot V_{\text{max}}(\gamma'). \quad (7)$$

Combining Eqs. (6) and (7), and letting $V_{\text{max}}(\gamma') = r_{\text{max}}/(1-\gamma')$, the error between the true value function

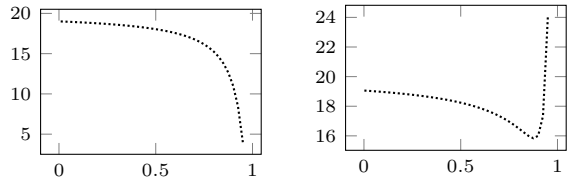


Fig. 1. Error bound $e_1(\gamma')$ (y-axes) as a function of lowered discount factor γ' (x-axes) for two different values of ϵ (0.01 and 0.06 in the left and right plots, respectively).

and approximate value function with lowered discount factor, $\|V_\gamma - U_{\gamma'}\|_\infty$, is bounded by

$$\begin{aligned} e_1(\gamma') &= \frac{\gamma - \gamma'}{(1-\gamma')(1-\gamma)} r_{\max} + \frac{\epsilon}{(1-\gamma')^2} r_{\max} \\ &\equiv f_1(\gamma') + \epsilon \cdot f_2(\gamma') \end{aligned} \tag{8}$$

Figure 1 illustrates the error bound $e_1(\gamma')$ with two different values of ϵ. Without loss of generality, $r_{\max}$ is scaled to 1. In addition, γ is set to 0.95. In Eq. (8), f_1 is a decreasing function while f_2 is increasing. When $\epsilon = 0.01$, i.e., the difference between the exact update and the approximate update is small, f_1 dominates f_2 and the approximate value function would not be improved with a lowered discount factor, as shown in Fig. 1 (left). This situation might happen, for example, if the observation model provides near-certain information of the environment. The POMDP would thus behave like an MDP, and the QMDP update is close to an exact update.

On the other hand, when the difference between an approximate update and the exact update becomes large, such as $\epsilon = 0.06$, the approximation error as the second term in Eq. (8) is comparable to the discount error. These two errors lead to a trade-off and the minimum occurs at a certain lowered discount factor, as shown in Fig. 1 (right). This is relevant, for instance, when the observation model is noisy and less informative. Since QMDP incorrectly assumes full observability, the difference in updates, ϵ, becomes larger. The effectiveness of lowering the discount factor while planning with different levels of observation noise will be demonstrated in a later section.

B. Sparse Reward Scenario

When the reward function is sparse, a tighter bound can be obtained. Assume $\vec{r} = \{r_1, r_2, r_3, ...\} \in \vec{R}$ is a sequence of rewards that an agent obtains in a trajectory, with $\vec{R}$ being the set of all possible reward trajectories. Let M be a positive integer such that

$$\begin{aligned} M = \min_{n \in \mathbb{N}_{\geq 0}, \vec{r} \in \vec{R}} [m \in \mathbb{N}_{>0} \mid r_{n-1} \neq 0, r_n = 0, \ldots, \\ r_{n+m-2} = 0, r_{n+m-1} \neq 0]. \end{aligned} \tag{9}$$

Here we assume the decision process starts at $t = 0$. For example, $\vec{r} = \{0, 0, 1, 0, 0, 1, \ldots\}$ corresponds to $M = 3$. If M is larger, then the reward is more sparse. In addition, $V_{\max}(\gamma) = r_{\max} \cdot \gamma^{M-1}/(1-\gamma^M)$. Therefore, by replacing

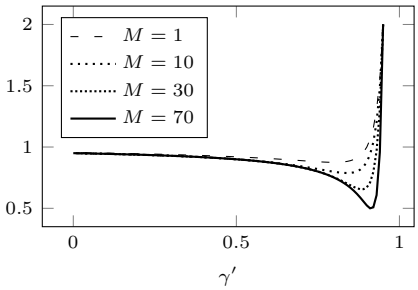


Fig. 2. Normalized error $\bar{e}_M$ as a function of lowered discount factor γ' with respect to different values of M in Eq. (9).

$V_{\max}$ in Eq. (6) and $V_{\max}(\gamma')$ in Eq. (7) by this formula, a new error bound is obtained:

$$\frac{(\gamma - \gamma')\gamma^{M-1} r_{\max}}{(1-\gamma')(1-\gamma^M)} + \frac{\epsilon \cdot {\gamma'}^{M-1} r_{\max}}{(1-\gamma')(1-{\gamma'}^M)}. \tag{10}$$

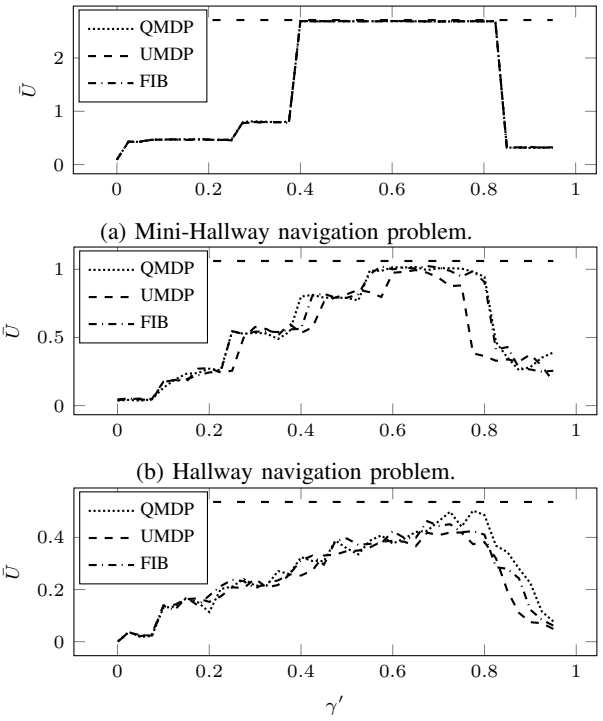
When $M = 1$, Eq. (10) is just Eq. (8). When M increases, the error bound becomes tighter. To demonstrate this, curves of error bound with respect to different values of M are plotted. However, since each M corresponds to different POMDP problems, normalization is necessary before making comparisons. Let $e_M(\gamma')$ be the error bound in Eq. (10). Define the normalized error as $\bar{e}_M = e_M / V_{\max}$, where $V_{\max} = r_{\max} \cdot \gamma^{M-1}/(1-\gamma^M)$.

Figure 2 plots the normalized error $\bar{e}_M$ as a function of γ' with respect to four values of M. Note that in the figure, $\epsilon = 0.1$, $\gamma = 0.95$, and $r_{\max} = 1.0$. Since the error bound is normalized, all curves coincide at $\gamma' = 0$ and $\gamma' = \gamma$, the two ends of curves. As M increases, the minimal values of each curves decreases and thus the error bound becomes tighter. In addition, the minimizer for each curve γ'^*_M slightly increases toward γ. These properties, such as the tighter bound and the shift of the minimizers, will be later supported with experiments.

IV. Experiments

The state-space planners approximate POMDP value functions inaccurately by associating each action with only one alpha vector. This approximation introduces error in accounting for future events in value iteration. The longer the effective planning horizon, the more error introduced by the model inaccuracy. This section explores the effect of reducing discount factors, thereby shortening the effective horizons in state-space planners, to trade off between long-term rewards and error introduced by approximation. Note that the loss of long-term rewards is related to $\|V_\gamma - V_{\gamma'}\|_\infty$ and the error introduced by approximation is $\|V_{\gamma'} - U_{\gamma'}\|_\infty$ in Eq. (5).

In this section, POMDP policies are evaluated by their discounted returns, $\bar{U} = \sum_{t=0}^{\infty} \gamma^t r_t$, where r_t is the immediate reward obtained at time t and $\gamma \leq 1$ is the true discount factor. QMDP, UMDP, FIB policies are generated using a lowered discount factor $\gamma' \leq \gamma$. *The resulting policies represented as alpha vectors are still evaluated by the*

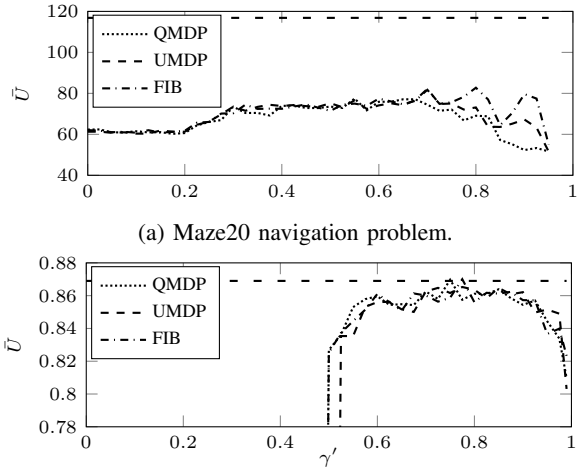


(a) Mini-Hallway navigation problem.

(b) Hallway navigation problem.

(c) Hallway2 navigation problem.

Fig. 3. Hallway Navigation Problems: (a) Mini-Hallway. (b) Hallway. (c) Hallway2.



(a) Maze20 navigation problem.

(b) MIT navigation problem.

Fig. 4. Two Navigation Problems.

discounted returns $\bar{U}$ with the true discount factor γ. All the files for the experiments were obtained from `pomdp.org`. Note that all figures in this experiment section share the same axis labels; the horizontal axis is the lowered discount factor γ' and vertical axis is the discounted return $\bar{U}$.

A. Benchmark Problems

The effect of lowering the discount factor while planning is tested on seven benchmark problems. For each benchmark problem, the relation between the lowered planning discount factor γ' and the discounted return $\bar{U}$ evaluated with the true discount factor γ is estimated by averaging over 1000 Monte Carlo evaluations, except for the large navigation problems which require 2000 Monte Carlo evaluations.

a) Hallway navigation problems: Figure 3 shows the results on three hallway navigation problems [5]. For example, in Fig. 3a, QMDP planned with discount factor 0.95 has discounted return 0.318. When the planning discount factor is reduced to 0.8, the performance of QMDP is significantly improved, with discounted return 2.691. Note that the horizontal line is the discounted return of a near-optimal SARSOP policy. In the Mini-Hallway navigation, QMDP, UMDP, and FIB have nearly the same performance, and can compete with SARSOP by reducing the planning discount factor γ'. Similar results can be found in the Hallway (Fig. 3b) and Hallway2 (Fig. 3c) navigation problems.

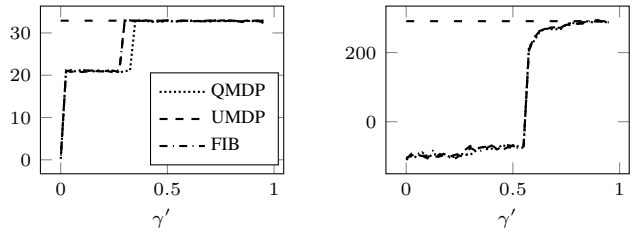
b) Computation time: State-space planners are known for their low runtime. For instance, in the Hallway2 navigation problem, which has the largest state space among the three problems, the convergence time for QMDP is less than one second. Lowering the discount factor can further accelerate convergence. QMDP with $\gamma = 0.8$ converges within 0.2 seconds. On the other hand, SARSOP requires several hours to converge.

c) Other Navigation Problems: Figure 4 shows the results on two more difficult navigation problems. For consistency with the previous problems, the true discount factor γ is set to 0.95. Figure 4a shows results on the Maze20 problem [3] and Fig. 4b on the MIT navigation problem [14]. The former is challenging for state-space planners since it has information-gathering actions and non-zero rewards for each position in the maze. The latter is also challenging since it has a larger state space. Again, reducing the planning discount factor can improve the performance in terms of discounted return.

d) Two other benchmark problems: Figure 5 shows the effect of lowering the discount factor in the Shuttle Docking problem [15] and the Network Problem [16]. For consistency, the true discount factor is set to 0.95. In the two problems, QMDP, UMDP, and FIB with the true discount factor already provides near-optimal results, comparable to the performance of SARSOP as indicated by the dashed horizontal lines. Thus, no improvement is observed if the discount factor is lowered while planning, which corresponds to the case of Eq. (8) with low ϵ. However, as will be shown, increased uncertainty in the observation model can result in a significant decrease in the performance of QMDP, UMDP, and FIB, and the improvement from lowering the discount factor can appear again.

B. Model Inaccuracy due to Observation Noise

Two experiments are proposed to illustrate the relationship between a lowered discount factor and model inaccuracy. A higher approximation error is introduced due to model inaccuracy, more improvement from lowering planning discount factor can be observed. Each experiment is based on a benchmark problem: the Mini-Hallway navigation problem [5] and



(a) Shuttle Docking Problem (b) Network Problem

Fig. 5. Lowered discount factor in on two benchmark problems. Since these benchmark problems are relatively easy, the state-space planners can reach near-optimal solutions.

the Shuttle Docking problem [15]. For each problem, the state space S, action space A, transition function T, and initial belief distribution follow the original experimental setup. However, the observation space Ω is set to be the same as S, and the observation function O is

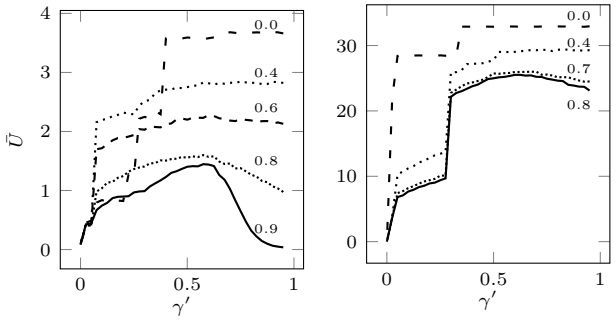
$$O^o_{s',a} = \begin{cases} 1-\delta, & \text{if } o = s' \\ \frac{\delta}{|\Omega|-1}, & \text{otherwise} \end{cases} \tag{11}$$

If $\delta = 0$, these experiments are simply MDPs, and thus ϵ in Eq. (8) by QMDP or FIB is zero. If $\delta \neq 0$, then these experiments are POMDPs. In addition, as δ increases, the model inaccuracy from the state-space planners such as QMDP and FIB also increases, and the corresponding ϵ are also magnified.

Figure 6a shows how the relation between the lowered discount factor γ' and discounted return $\bar{U}$ varies with respect to δ on the Mini-Hallway navigation problem. Here, the state-space planner is FIB and $\bar{U}$ is estimated by averaging over 1000 Monte Carlo evaluations. When $\delta = 0$, the system is an MDP, and the optimal result is obtained by $\gamma' = \gamma = 0.95$. However, if δ increases, the performance of FIB degrades because it approximates the problem poorly and ϵ in Eq. (5) also increases. The improvement from lowering the discount factor thus becomes more obvious. Especially, for $\delta = 0.9$, with strong uncertainty from observation model, FIB with a lowered discount factor $\gamma' = 0.6$ can reach the performance of SARSOP. Figure 6b reveals the same result on the Shuttle Docking problem. The original Shuttle Docking problem has no improvement when the planning discount factor is lowered, as in Fig. 5a, since FIB already reaches the near-optimal result. However, when the observation model becomes more noisy, FIB performance degrades and the lowered discount factor outperforms.

C. Reward Sparsity

The purpose of lowering the discount factor while planning is to reduce errors in the value function. As shown in the theoretical analysis and Fig. 2, the error is bounded more tightly when the reward is sparse. If the reward function is sparse, a long effective horizon is required to see the future rewards and solving the POMDP problem by value iteration would introduce more errors. In this case, the effectiveness of lowering the discount factor would be more significant.



(a) Mini-Hallway (redesigned) (b) Shuttle (redesigned)

Fig. 6. Results on redesigned problems. Labels on curves are the values of δ in Eq. (11).

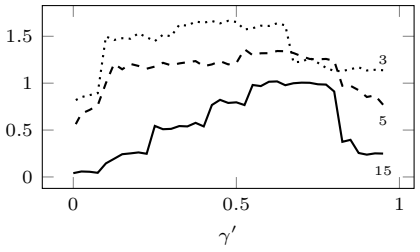


Fig. 7. Lowered discount factor in the Hallway navigation problem with different values of M in Eq. (10).

For example, in the hallway navigation problems, an agent is only rewarded when it arrives at the goal position and thus long horizons in the mazes are required. As a result, the improvement from lowering the discount factor is significant, as shown in Figs. 3a to 3c.

To further demonstrate this property, two experiments based on the Mini-Hallway and Hallway navigation problems are introduced to connect the improvement from lowering the discount factor with the sparsity of reward. In the Hallway navigation problem, only the goal state rewards agents by $+1.0$, which means that the corresponding M in Eq. (10) is approximately 15. For the two new scenarios in Fig. 7, an agent receives $+0.1$ rewards when it finishes every one-third and every one-fifth of the maze, respectively. That is to say, the corresponding values of M are 5 and 3.

Figure 7 summarizes the results with policies generated by FIB. In the original Hallway problem, there is only one reward in the maze and the improvement from lowering the discount factor is 0.76. When there are landmark rewards at every one-third and every one-fifth of the maze, the improvement is $+0.59$ and $+0.51$, respectively. This confirms the theoretical analysis that the improvement from lowering the discount factor is most prominent when the reward function is sparse. Furthermore, the optimal lowered discount factor γ^* is also shifted toward the true γ as M increases. This shift is also predicted by the theoretical analysis.

The sparsity of the reward function is related to the magnitude of landmark rewards in the maze. As the value increases, the landmark rewards can reduce the required horizon. This generalization of sparse reward is tested on the

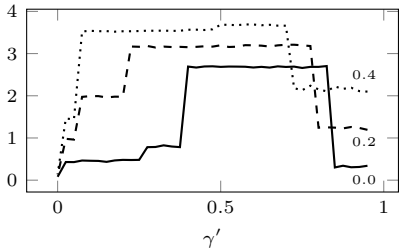


Fig. 8. Lowered discount factor in the Mini-Hallway navigation problem with different values of r_{mid}.

Mini-Hallway problem. In the original scenario, an agent is only rewarded by $+1.0$ when it reaches the goal position s_{13}. To reduce sparsity, an landmark reward r_{mid} is introduced halfway through the maze at state s_6. As the value of r_{mid} increases, the sparsity decreases.

Figure 8 summarizes the effectiveness of lowering the discount factor in planning versus the different values of r_{mid} for the Mini-Hallway problem. As the value of r_{mid} increases, the improvement from lowering the discount factor is less significant. In the original Mini-Hallway problem (with $r_{\text{mid}} = 0$), the improvement is $+2.3$. When r_{mid} is set to 0.2 and 0.4, the improvements are $+2.0$ and $+1.6$, respectively. Figure 8 clearly shows that the improvement is most significant when the function is sparse. Furthermore, the shift of the optimal lowered discount factor γ^* is observed: as landmark value decreases, the reward is more sparse, and γ^* approaches the true discount factor γ.

V. Discussion and Conclusion

The following two papers are related to our work. Petrik and Scherrer [9] first showed that lowering the discount factor can improve performance in the context of fully observable MDPs. It has been proven that a lowered discount factor can reduce loss when the model is inaccurate. Jiang et al. [10] explored a similar phenomenon in reinforcement learning with fully observable states (in contrast to our partially observable and model-based approach). It was shown that the policy found using a shorter effective horizon, i.e., lowered discount factor, can actually be better than a policy found with the true discount factor. The authors connect model complexity with the planning horizon as an analogy to over-fitting in supervised learning. The longer the horizon, the greater the risk of overfitting.

In this paper, we concentrated on exploring the phenomenon in POMDP domains by providing theoretical results and testing on benchmark problems. Several sets of experiments were performed to confirm the theoretical predictions. However, we did not specify methods to determine the lowered discount factor. In fact, the dependence of POMDP performance on a lowered discount factor may be non-trivial. A practical approach is to search between $\gamma/2$ and γ for γ' with the assumption that the improvement is approximately unimodal. This is computationally feasible since the discount factor is a one-dimensional variable and the computational cost of QMDP, UMDP, and FIB is low. QMDP and UMDP only require $O(|S|^2|A|^2)$ operations per iteration. Moreover, as the planning discount factor is reduced, the number of iterations required for convergence is also reduced.

Finally, this phenomenon is potentially applicable to point-based methods as well. When the number of belief points and their associated alpha vectors is small, point-based methods would also have inaccuracies in their approximation, as is the case for the state-space planners.

References

[1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[2] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

[3] M. Hauskrecht, "Value-function approximations for partially observable Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 13, pp. 33–94, 2000.

[4] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and Systems*, vol. 6, 2008, pp. 65–72.

[5] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning (ICML)*, 1995, pp. 362–370.

[6] N. A. Vien and M. Toussaint, "POMDP manipulation via trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 242–249.

[7] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 244–264, 2016.

[8] S. Pellegrinelli, H. Admoni, S. Javdani, and S. Srinivasa, "Human-robot shared workspace collaboration via hindsight optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 831–838.

[9] M. Petrik and B. Scherrer, "Biasing approximate dynamic programming with a lower discount factor," in *Advances in Neural Information Processing Systems (NIPS)*, 2009, pp. 1265–1272.

[10] N. Jiang, A. Kulesza, S. Singh, and R. Lewis, "The dependence of effective planning horizon on model accuracy," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015, pp. 1181–1189.

[11] N. Jiang, S. Singh, and A. Tewari, "On structural properties of MDPs that bound loss due to shallow planning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[12] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, 2013.

[13] M. T. J. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[14] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 1996, pp. 963–972.

[15] L. Chrisman, "Reinforcement learning with perceptual aliasing: The perceptual distinctions approach," in *National Conference on Artificial Intelligence (AAAI)*, 1992, pp. 183–188.

[16] M. L. Littman, "Algorithms for sequential decision making," Ph.D. dissertation, Brown University, 1996.