PLEASE: Palm Leaf Search for POMDPs with Large Observation Spaces

Zongzhang Zhang[a,b,c], **David Hsu**[c], **Wee Sun Lee**[c], **Zhan Wei Lim**[c], **Aijun Bai**[d]
[a]School of Computer Science and Technology, Soochow University, Suzhou 215006, P.R. China
[b]Collaborative Innovation Center of Novel Software Technology and Industrialization, Jiangsu, P.R. China
[c]Department of Computer Science, National University of Singapore, Singapore 117417, Singapore
[d]School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230027, P.R. China
zzzhang@suda.edu.cn, {dyhsu, leews, limzhanw}@comp.nus.edu.sg, baj@mail.ustc.edu.cn

Abstract

This paper provides a novel POMDP planning method, called *Palm LEAf SEarch* (PLEASE), which allows the selection of more than one outcome when their potential impacts are close to the highest one during its forward exploration. Compared with existing trial-based algorithms, PLEASE can save considerable time to propagate the bound improvements of beliefs in deep levels of the search tree to the root belief because of fewer backup operations. Experiments showed that PLEASE scales up SARSOP, one of the fastest algorithms, by orders of magnitude on some POMDP tasks with large observation spaces.

Introduction

Partially Observable Markov Decision Process (POMDP) provides a rich mathematical framework for agent reasoning and planning in uncertain environments. In the past decade, point-based algorithms have made impressive progress in handling large POMDPs. Several representative algorithms among them, such as HSVI2, GapMin and SARSOP (Kurniawati, Hsu, and Lee 2008), do a *trial-based search* in their sampling strategies, where the current bounds of the optimal value function V^* are used to select a *single path* from the initial belief b_0 to one leaf belief during each trial.

This paper provides a new method called *Palm LEAf SEarch* (PLEASE). We use a simple example to explain the motivation behind the method. Assume that, for a POMDP, n child beliefs of a belief at level $d-1$, denoted b_{d-1}, need to perform point-based value backups to propagate their bound improvements to the initial belief b_0 to guarantee to find a near optimal policy at b_0. For each trial, trial-based algorithms need to forward search one of b_{d-1}'s n child beliefs so as to propagate its improvement to b_0. Since each trial length is $d+1$, totally, there are $n \times (d+1)$ beliefs that need to perform backups. The PLEASE method allows the selection of *more than one* outcome during its forward exploration phase when their potential impacts are close to the highest one. Instead of repeatedly performing backups on b_i, where $i = 0, \cdots, d-1$, n times, PLEASE can perform them *once*. As a result, there are only $n + d$ beliefs that need to perform backups in PLEASE. When d or n becomes larger, $\frac{d(n-1)}{n+d}\big(= \frac{n(d+1)-(n+d)}{n+d}\big)$ increases. This suggests that PLEASE is more attractive when tackling a POMDP with *large* observation size (probably large n) and that needs to search *deeply* (large d) to find a near optimal solution.

Algorithm 1 EXPLORE(b, d_b, ϵ) in PLEASE.

1: **if** b's gap termination condition == true **then**
2: **return** ;
3: **end if**
4: $a^* = \arg\max_{a \in A} Q^U(b, a)$;
5: $z^* = \arg\max_{z \in Z}[Pr(z|b, a^*) \cdot \text{excess}(\tau(b, a^*, z), d_b + 1, \epsilon)]$;
6: **for** $z \in Z$ **do**
7: **if** $Pr(z|b, a^*) \cdot \text{excess}(\tau(b, a^*, z), d_b + 1, \epsilon) \geq \zeta \cdot Pr(z^*|b, a^*) \cdot \text{excess}(\tau(b, a^*, z^*), d_b + 1, \epsilon)$ **then**
8: EXPLORE$(\tau(b, a^*, z), d_b + 1, \epsilon)$;
9: **end if**
10: **end for**
11: Perform a backup operation of bounds at belief b;

Palm Leaf Search

Compared with SARSOP, the EXPLORE procedure in PLEASE (see Algorithm 1) adds a for loop in Lines 6∼10. This allows Algorithm 1 to search towards more than one outcome in its forward exploration phase when their potential impacts are close to the highest one. How to use the heuristics of beliefs to select promising child beliefs to make PLEASE do best is *not* trivial. Here, we control the palm leaf search at beliefs by setting ζ online (Line 7 in Algorithm 1). We define it as a function of b and θ, called $\zeta(b, \theta)$, where θ is changeable over time and independent of b. We omit the definition of symbols in Algorithm 1. For more details, please refer to the full paper (Zhang et al. 2015).

To make PLEASE work well in all POMDP cases, we give users an input constant C in exploiting the prior knowledge. The constant C is defined as the *desired* ratio of #PLEASE - #SARSOP to #SARSOP, where #PLEASE is the total number of backups in PLEASE, and #SARSOP is the total number of backups on the paths selected by SARSOP's action and outcome selection strategies and its belief's gap termination condition. In contrast to SARSOP,

$C > 0$ reduces the number of backups to propagate the bound improvements of selected leaf nodes to the root node. Please see (Zhang et al. 2015) for a variant of PLEASE (called PLEASE-Z) with a different definition of C.

We let PLEASE control θ online so that it obtains an ideal value of θ over time to make the actual ratio of #PLEASE to #SARSOP close to $C+1$. Specifically, we define θ by using the following rule:

$$\theta = \begin{cases} \min\{\theta + \Delta, 1\} & \text{if } \frac{\#\text{PLEASE}}{\#\text{SARSOP}} \geq C+1, \\ \max\{\theta - \Delta, \theta_l\} & \text{otherwise.} \end{cases}$$

Here, θ is set to $\theta_0 = 1$ in the beginning of PLEASE, $\theta_l = 0.8$, and $\Delta = 0.01$ in this paper. PLEASE adjusts the value of θ in the beginning of each forward exploration phase from the initial belief b_0 in the PLEASE method.

PLEASE defines $\zeta(b, \theta) = \sqrt[\text{dis}(b,p^*)+1]{\theta}$ by using θ and b's heuristic information to achieve the goal of giving more time to do palm leaf search in promising beliefs with deep levels. Here, p^* represents the path generated by SARSOP's action and outcome selection strategies and its belief's gap termination condition, and the distance from b to p^*, denoted $\text{dis}(b, p^*)$, as the number of beliefs that b needs to go through to arrive in p^*. Such a formula guarantees that PLEASE does more aggressive palm leaf search around the current best path p^* in each forward exploration phase starting at b_0.

Essentially, palm leaf search can be viewed as a kind of complete anytime best-first beam search in POMDPs. Similar techniques of tree-trials have been used in Monte-Carlo tree search, such as df-UCT by Yoshizoe et al. (2011). At each step, the complexity of each PLEASE exploration step is at least the time and space complexity of each SARSOP step. However, its conservative theoretical time bound of finding an ϵ-optimal policy is verifiable to be not worse than the original trial-based algorithm.

Experiments

We use Tag(55) and Two-Robot Tag, the two POMDP problems with large observation spaces, to study the effects of varying the tuning parameter C. Empirical results show that $C = 4$ and 10 are good candidates for the two problems, respectively. We use $C = m \log_{10} |Z|$, inspired by the fact that C should be larger when $|Z|$ increases, as a simple formula to set the input constant automatically. Here, $m = 3.22$ is fitted by using the least squares method, which uses $C = 4$, $|Z| = 56$ from the Tag(55) problem and $C = 10$, $|Z| = 625$ from the Two-Robot Tag problem as the training set.

Table 1 compares PLEASE with SARSOP on 8 benchmark problems in terms of the gap between bounds at b_0 ($V^U(b_0) - V^L(b_0)$), the lower bound $V^L(b_0)$, the upper bound $V^U(b_0)$ and the total running time (in seconds). On most of these problems this table shows that PLEASE is substantially faster than SARSOP by orders of magnitude.

There are three other observations obtained from more detailed data in our experiments. First, the performance of HSVI2 on our test problems is worse than SARSOP in most cases, and GapMin variants are not efficient in tackling large problems. Second, compared with SARSOP, PLEASE needed much fewer α-vectors to get the same gaps on test problems. For example, when the gap is 3.43 on HomeCare, the number of α-vectors in PLEASE is 7,947, while 30,633 in SARSOP. Third, when the gaps were the same, SARSOP and PLEASE's expected total rewards appeared to be similar with each other. Specifically, expected total reward was -9.73 ± 0.12 on Tag(55) when the gap was 5.21; the reward was -11.58 ± 0.12 on Two-Robot Tag(24) when the gap was 7.43; and the expected reward was 17.03 ± 0.14 on HomeCare when the gap was 3.43.

Table 1: Comparison of SARSOP and PLEASE.

Algorithm	Gap	$V^L(b_0)$	$V^U(b_0)$	Time
Tag(55) ($\|S\| = 3,080, \|A\| = 5, \|Z\| = 56$)				
SARSOP	5.21	-9.89	-4.68	9,836
PLEASE(C=4)	5.21	-9.88	-4.67	1,527
Two-Robot Tag ($\|S\| = 14,400, \|A\| = 25, \|Z\| = 625$)				
SARSOP	6.56	-12.28	-5.72	99,972
PLEASE(C=10)	6.53	-11.88	-5.35	4,174
Tag(85) ($\|S\| = 7,310, \|A\| = 5, \|Z\| = 86$)				
SARSOP	7.02	-12.45	-5.43	9,995
PLEASE(C=6.23)	7.02	-12.18	-5.16	583
Tag(102) ($\|S\| = 10,506, \|A\| = 5, \|Z\| = 103$)				
SARSOP	7.68	-13.56	-5.88	9,993
PLEASE(C=6.48)	7.66	-13.15	-5.49	365
Hallway ($\|S\| = 60, \|A\| = 5, \|Z\| = 21$)				
SARSOP	0.18	1.01	1.19	9,995
PLEASE(C=4.26)	0.18	1.00	1.18	115
Hallway2 ($\|S\| = 92, \|A\| = 5, \|Z\| = 17$)				
SARSOP	0.46	0.42	0.88	9,990
PLEASE(C=3.96)	0.46	0.40	0.86	426
FieldVisionRockSample_5_5 ($\|S\| = 801, \|A\| = 5, \|Z\| = 32$)				
SARSOP	0.47	23.27	23.74	9,764
PLEASE(C=4.85)	0.47	23.28	23.75	3,585
HomeCare ($\|S\| = 5,408, \|A\| = 9, \|Z\| = 928$)				
SARSOP	2.98	16.77	19.75	99,686
PLEASE(C=9.56)	2.96	16.77	19.73	3,706

Future Topics One topic is to enrich the theoretical analysis of the heuristic for observation selection in the current PLEASE method. Another interesting topic is how to use more heuristic information of beliefs (e.g., the depth information of beliefs) in defining the threshold function $\zeta(b, \theta)$.

Acknowledgements

This work was supported in part by MoE AcRF grant 2010-T2-2-071 and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Kurniawati, H.; Hsu, D.; and Lee, W. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *RSS*.

Yoshizoe, K.; Kishimoto, A.; Kaneko, T.; Yoshimoto, H.; and Ishikawa, Y. 2011. Scalable distributed Monte-Carlo tree search. In *SoCS*, 180–187.

Zhang, Z.; Hsu, D.; Lee, W.; Lim, Z.; and Bai, A. 2015. PLEASE: Palm leaf search for POMDPs with large observation spaces. In *ICAPS*, 249–257.