LB-DESPOT: Efficient Online POMDP Planning Considering Lower Bound in Action Selection* (Student Abstract)

Chenyang Wu [1], Rui Kong [1], Guoyu Yang [1], Xianghan Kong [1], Zongzhang Zhang [1], Yang Yu [1], Dong Li [2], Wulong Liu [2]

[1] National Key Lab for Novel Software Technology, Nanjing University, China
[2] Noah's Ark Lab, Huawei Company
{wucy, yanggy, kongxh, zhangzz, yuy}@lamda.nju.edu.cn, {lidong106, liuwulong}@huawei.com

Abstract

Partially observable Markov decision process (POMDP) is an extension to MDP. It handles the state uncertainty by specifying the probability of getting a particular observation given the current state. DESPOT is one of the most popular scalable online planning algorithms for POMDPs, which manages to significantly reduce the size of the decision tree while deriving a near-optimal policy by considering only K scenarios. Nevertheless, there is a gap in action selection criteria between planning and execution in DESPOT. During the planning stage, it keeps choosing the action with the highest upper bound, whereas when the planning ends, the action with the highest lower bound is chosen for execution. Here, we propose LB-DESPOT to alleviate this issue, which utilizes the lower bound in selecting an action branch to expand. Empirically, our method has attained better performance than DESPOT and POMCP, which is another state-of-the-art, on several challenging POMDP benchmark tasks.

Introduction

The partially observable Markov decision process (POMDP) provides a mathematical framework for modeling sequential decision-making problems with uncertainty in state transition and observation (Kochenderfer 2015). Due to the overwhelming computational complexity of POMDPs, online algorithms are indispensable, especially when solving large-scale problems. Because rather than solving for a globally optimal policy, they only try to find an acceptable local policy for the current belief, and therefore significantly relieves the computational overload. DESPOT (Ye et al. 2017) is a state-of-the-art online planing algorithm, which derives its name from the fact that it solves for a near-optimal policy by constructing a determinized sparse partially observable tree. One central idea in DESPOT is using branch-and-bound to accelerate the searching process and find a near-optimal action for execution. It maintains upper and lower bounds of state-action value for each action. During the planning, DESPOT selects the action with the highest upper bound for expansion. When the planning ends, the action with the highest lower bound is chosen for execution.

	Laser Tag	Drone	Roomba
DESPOT	-10.86 ± 0.06	10.62 ± 0.05	-2.12 ± 0.01
POMCP	-16.34 ± 0.06	10.42 ± 0.04	-2.30 ± 0.01
Value LB	$-\ 9.79 \pm 0.06$	10.65 ± 0.05	-2.09 ± 0.01
Prob LB	-10.80 ± 0.06	11.08 ± 0.05	-2.06 ± 0.01
Rank LB	$-\ 9.44 \pm 0.06$	11.08 ± 0.05	-1.97 ± 0.01

Note: Value (Prob, Rank) LB=Value (Probability, Ranking)-based LB-DESPOT

Table 1: Performance comparison

Given infinite planning time, the upper bound based search strategy is guaranteed to find an optimal action; nonetheless, the planning time is always limited in online planning. When the planning time is limited, DESPOT cannot fully explore branches with a potential high lower bound, since it always expands the action branch with the highest upper bound, which eventually leads to an inferior action. This paper proposes a new method, LB-DESPOT, to improve the action selection in DESPOT. LB-DESPOT takes advantage of not only the upper bound but also the lower bound to select an action branch during the exploration and therefore clears the gap of selection criteria between planning and execution, leading to better performance.

Methods

In the section, we present three different implementations of LB-DESPOT while providing empirical analysis for each.

Value-based LB-DESPOT is a direct method which simply chooses an action branch according to the weighted sum of the lower and upper bounds, i.e.,

$$a^* = \arg\max_{a \in \mathcal{A}} \left(\beta \cdot \ell(b,a) + \mu(b,a) \right), \tag{1}$$

where $\ell(b,a)$ and $\mu(b,a)$ are the lower and upper bounds of the action a at belief b respectively, and β is a coefficient for adjusting the participation of the lower bound. However, this method suffers from the problem of instability, as a β shared across all belief nodes will not suit all of them. A reasonable choice of β should make sure $\beta \cdot \ell(b,a)$ and $\mu(b,a)$ being comparable, i.e., in the same order of magnitude. When the orders of magnitude of the lower and upper bounds are similar, a β close to 1 might suffice. Yet when the upper bound

*Corresponding author: Zongzhang Zhang. This work is in part supported by the National Natural Science Foundation of China (No. 61876119) and Huawei Noah's Ark Lab (No. HF2019105005).

	Laser Tag	RS(7,8)	RS(11,11)	RS(15,15)	Drone Surveillance	Roomba
DESPOT	-12.95 ± 0.07	6.73 ± 0.06	5.46 ± 0.05	4.27 ± 0.05	10.36 ± 0.05	-2.36 ± 0.01
LB-DESPOT	-11.88 ± 0.06	7.77 ± 0.06	7.88 ± 0.06	6.95 ± 0.06	10.47 ± 0.05	-2.37 ± 0.04

Note: RS(n,m) stands for the Rock Sample with $n \times n$ map and m rocks.

Table 2: Empirical results when a shoddy lower bound generated by the rollout of the random policy is given

is several orders of magnitude greater than the lower bound, the value of a suitable β could change significantly. Considering that it is common for the lower or upper bound being close to 0, which means the order of magnitude could vary significantly, the performance of this method can be unpredictable.

Probability-based LB-DESPOT chooses an action according to the lower bound solely with a probability β:

$$a^* = \xi > \beta \, ? \arg\max_{a \in \mathcal{A}} \mu(b, a) : \arg\max_{a \in \mathcal{A}} \ell(b, a), \qquad (2)$$

where $\xi \sim \mathrm{Uniform}(0, 1)$. This method does not suffer the instability numerically, but features the inborn randomness, behaving unpredictably in practice. So it is less preferable.

Ranking-based LB-DESPOT is the final choice of implementation for LB-DESPOT. It exploits the ranking of bounds to determine the action branch to expand. Firstly, it ranks all available action branches with respect to the lower and upper bounds, respectively. Then, it chooses an action branch to expand in agreement with the comprehensive rankings, i.e.,

$$a^* = \arg\max_{a \in \mathcal{A}} \left(\beta \cdot \mathrm{rank}^{\ell}_{b,a} + \mathrm{rank}^{\mu}_{b,a} \right), \qquad (3)$$

where $\mathrm{rank}^{\mu}_{b,a}$ and $\mathrm{rank}^{\ell}_{b,a}$ are the descending rankings of the action a in terms of upper and lower bounds of belief b, respectively. Specifically, in $\mathrm{rank}^{\ell}_{b,a}$, the action with the highest lower bound is set to rank last so that it is precluded from being chosen. Because choosing the action with the highest lower bound continually will sometimes cause the algorithm to get stuck in a local optimum.

Experiments

In this section, we experiment on four challenging POMDP benchmark experiments: Laser Tag ($|\mathcal{S}| = 4,830$, $|\mathcal{A}| = 5$, $|\mathcal{Z}| \approx 1.5 \times 10^6$), Drone Surveillance ($|\mathcal{S}| = 625$, $|\mathcal{A}| = 5$, $|\mathcal{Z}| = 10$), Roomba ($|\mathcal{S}| = \infty$, $|\mathcal{A}| = 6$, $|\mathcal{Z}| = 11$), and Rock Sample ($|\mathcal{S}|$ and $|\mathcal{A}|$ vary in different settings, $|\mathcal{Z}| = 3$). Please see (Ye et al. 2017) for detailed description of these problems. Here, $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{Z}$ represent state, action, and observation spaces in a POMDP model, respectively, and $|\cdot|$ stands for the cardinality of a set. We compare the performance of LB-DESPOT with DESPOT and POMCP (Silver and Veness 2010). For each algorithm, the best heuristic and hyperparameter set are found beforehand. Specifically, we implement both the upper bound and default policy for DESPOT and LB-DESPOT in each problem. For POMCP, the rollout policy and the initializer of N_{init} and V_{init} are provided. The regularization parameter λ is selected from the set $\{0.0, 0.01, 0.1, 1.0, 10.0\}$, and the number of scenarios K from $\{30, 100, 300, 1000, 3000\}$. The best lower bound participation parameter β for LB-DESPOT is selected from $\{0.0, 0.1, 0.3, 1.0\}$. All the parameter selections are conducted in a different experimental setting from the one used for testing. During testing, experiments are conducted 100 episodes per problem, with the max step being 100 steps, and algorithms are allocated 1 second per step for planning. In the tables, the average discounted returns are shown along with the corresponding standard errors of mean (SEM) in the form of $\mathrm{Return} \pm \mathrm{SEM}$.

As shown in Table 1, all three implementations of LB-DESPOT outperform DESPOT and POMCP. Among them, the ranking-based LB-DESPOT performs best; hence, it is chosen as the final implementation. The data on Rock Sample is not presented, because in Rock Sample, when a good lower bound is given, any participation of lower bound in action selection will incur degeneration of performance, of which we are still working on a satisfactory explanation.

Furthermore, we conduct another set of experiments to show that the greedy search strategy of LB-DESPOT works even when a shoddy initial lower bound is given. The experimental results are demonstrated in Table 2, where the initial lower bounds of DESPOT and LB-DESPOT are set as the rollout of the random policy. The results show that the greedy search strategy can work independent of the lower bound and provide an improvement over DESPOT, though a high-quality lower bound is always beneficial.

Future Work

One direction for our future work is to adjust the β dynamically according to the remaining time of planning. Since choosing an action according to lower bounds is a greedy way of searching, it might be wiser if the lower bound participation is adjusted according to the remaining time of planning. The lesser the remaining time is, the greater the participation should be.

Another promising direction is to improve the expansion of the observation branch. Our focus now is on the method, PLEASE, proposed in Zhang et al. (2015), which allows the selection of more than one observation branch when their potential impacts are close.

References

Kochenderfer, M. J. 2015. *Decision Making Under Uncertainty: Theory and Application.* MIT press.

Silver, D.; and Veness, J. 2010. Monte-Carlo planning in large POMDPs. In *NIPS*, 2164–2172.

Ye, N.; Somani, A.; Hsu, D.; and Lee, W. S. 2017. DESPOT: Online POMDP planning with regularization. *JAIR* 58: 231–266.

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