软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(7):1589−1600 [doi: 10.3724/SP.J.1001.2013.04318]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

杂合启发式在线 POMDP 规划*

章宗长, 陈小平

(中国科学技术大学 计算机科学与技术学院,安徽 合肥 230027)

通讯作者: 章宗长, E-mail: zzz@mail.ustc.edu.cn

摘 要: 许多不确定环境下的自主机器人规划任务都可以用部分可观察的马氏决策过程(partially observable Markov decision process,简称 POMDP)建模.尽管研究者们在近似求解技术的设计方面已经取得了显著的进展,开发高效的 POMDP 规划算法依然是一个具有挑战性的问题.以前的研究结果表明:在线规划方法能够高效地处理大规模的 POMDP 问题,因而是一类具有研究前景的近似求解方法.这归因于它们采取的是"按需"作决策而不是预前对整个状态空间作决策的方式.旨在通过设计一个新颖的杂合启发式函数来进一步加速 POMDP 在线规划过程,该函数能够充分利用现有算法里一些被忽略掉的启发式信息.实现了一个新的杂合启发式在线规划(hybrid heuristic online planning,简称 HHOP)算法.在一组 POMDP 基准问题上,HHOP 有明显优于现有在线启发式搜索算法的实验性能.

关键词: 部分可观察的马氏决策过程;在线规划;杂合启发法

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 章宗长,陈小平.杂合启发式在线 POMDP 规划.软件学报,2013,24(7):1589−1600. http://www.jos.org.cn/1000-9825/4318.htm

英文引用格式: Zhang ZZ, Chen XP. Hybrid heuristic online planning for POMDPs. Ruan Jian Xue Bao/ Journal of Software, 2013,24(7):1589−1600 (in Chinese). http://www.jos.org.cn/1000-9825/4318.htm

Hybrid Heuristic Online Planning for POMDPs

ZHANG Zong-Zhang, CHEN Xiao-Ping

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Corresponding author: ZHANG Zong-Zhang, E-mail: zzz@mail.ustc.edu.cn

Abstract: Lots of planning tasks of autonomous robots under uncertain environments can be modeled as a partially observable Markov decision processes (POMDPs). Although researchers have made impressive progress in designing approximation techniques, developing an efficient planning algorithm for POMDPs is still considered as a challenging problem. Previous research has indicated that online planning approaches are promising approximate methods for handling large-scale POMDP domains efficiently as they make decisions "on demand", instead of proactively for the entire state space. This paper aims to further speed up the POMDP online planning process by designing a novel hybrid heuristic function, which provides a feasible way to take full advantage of some ignored heuristics in current algorithms. The research implements a new method called hybrid heuristic online planning (HHOP). HHOP substantially outperformes state-of-the-art online heuristic search approaches on a suite of POMDP benchmark problems.

Key words: partially observable Markov decision process (POMDP); online planning; hybrid heuristics

部分可观察的马氏决策过程(partially observable Markov decision process,简称 POMDP)是一种通用的概率模型,它常被用于描述在行动效果和观察结果均不确定的环境中的规划问题[1].由于计算复杂度的原因,精确求解中、大规模的 POMDP 问题是不可能的[2].近 10 年来,研究者们开发了许多 POMDP 近似规划算法[3−7],并且把

* 基金项目: 国家自然科学基金(60745002, 61175057); 国家高技术研究发展计划(863)(2008AA01Z150)
收稿时间: 2012-02-13; 定稿时间: 2012-03-27

它们成功运用到了一些实际的机器人任务中,如人机口语对话[8]、RoboCup 四足机器人足球[9]、物体抓取[10]和服务机器人导航[11]等.POMDP 规划问题的难解源于“维数诅咒”和“历史诅咒”[4,12].维数诅咒是指在一个状态数为 n 的 POMDP 规划问题中,规划所需的计算须在 n 维状态空间中完成.表征维数诅咒的度量是状态数,缓解维数诅咒的方法是因子化状态表示法[7].历史诅咒是指不同历史的数目会随着规划步数的增加呈指数级别的增长.表征历史诅咒的度量是可达信念空间的δ-覆盖数[13],缓解历史诅咒的方法是启发式搜索法[3−6].最近的研究结果表明,在一组 POMDP 基准问题上,可达信念空间的δ-覆盖数是比状态数更好的、能够同时刻画 POMDP 规划和学习问题难度的度量[14].

根据规划方式的不同,POMDP 近似规划算法可分成两类,即离线规划算法和在线规划算法.离线规划算法由策略搜索和策略执行两个阶段组成,即先在策略搜索阶段用大量的预处理时间产生整个信念状态空间上的策略,然后用这个策略在策略执行阶段进行决策.由于离线规划算法的预处理时间可以被平均分摊到每次的任务中,因而它们适合求解重复的 POMDP 规划任务.而在线规划算法擅长处理紧急的或一次性的 POMDP 规划任务.它们不会分配大量的时间给预处理过程,而是在基于当前状态且时间有限的策略搜索步和策略执行步间迭代.在策略搜索步,在线规划算法重点在于计算好当前信念状态处的局部策略,而不是可以泛化到整个信念状态空间的全局策略.因而,较离线规划算法而言,在线规划算法有潜力在保证产生高回报的行动系列的同时,用于策略搜索步和策略执行步的总时间小得多.

正是由于在线规划算法的上述优点,它被认为是求解大规模 POMDP 问题的一类有研究前景的近似规划算法[15].然而,现有的在线规划算法在设计启发式搜索函数时忽略了一些重要的启发式信息,从而导致它们还不能做到足够高效.本文将讨论如何利用这部分启发式信息来设计一种更高效的杂合启发式搜索策略,以达到改进当前在线 POMDP 规划算法性能的目的.

我们的工作基于最优值函数的下界并没有在现有的在线规划算法[6]中得到充分利用这一事实.当前的在线规划通过保存最优值函数的下界和上界作为启发式信息来发现好的策略.决定算法整体性能的一个关键问题是如何利用最优值函数的上下界作为启发式信息来搜索近似最优策略.在以前的工作中,上界往往比下界更受欢迎,其原因在于:若朝着最高上界的行动分支搜索,算法能在有限时间里找到当前信念状态处的ε-最优行动[6].相比而言,下界更难被利用,如,总是沿着最高下界的行动分支搜索常会使搜索陷入到局部最优.然而,下界能够保证策略的质量,这一优点却是上界所不具有的.正因如此,在线规划算法返回的是最高下界而不是最高上界对应的行动作为当前的最优行动.

为了利用下界的这一优点,我们构造了一个基于下界的启发式搜索函数,其特点有:

(1) 能够把搜索导向到一组“有前景的”策略,这组策略与当前具有最高下界保证的策略相似但又有所不同,沿着这组策略搜索会较容易找到具有更好下界保证的策略;

(2) 能够在很大程度上避免搜索陷入局部最优.这些特点与当前流行的、基于上界的启发式搜索函数能够很好的优势互补.

我们用一种杂合的方法将这两个启发式搜索函数相结合,达到了充分利用上下界信息,同时避免它们各自缺陷的目的.结合后的算法被称为杂合启发式在线规划算法(hybrid heuristic online planning,简称 HHOP).在实验部分,我们全面比较了 HHOP 算法和一些现有的 POMDP 在线和离线规划算法在 9 个基准问题上的实验性能.实验结果表明,HHOP 算法使用的杂合启发式搜索策略比现有在线规划算法中的搜索策略更高效;较离线规划算法而言,HHOP 算法在保证得到高回报策略的同时,花费的规划总时间要小得多.

本文第 1 节介绍 POMDP 模型和在线规划算法的背景知识.第 2 节介绍 HHOP 算法.第 3 节实验分析 HHOP 算法的性能.第 4 节是结束语.

1 背景和相关工作

1.1 POMDP模型

POMDP 为建模主体在部分可观察的随机环境中的序贯决策问题提供了一个通用的数学模型.这里,我们

仅讨论的是离散的和带折扣因子的 POMDP 模型.它可以被形式化地定义为一个八元组$(S,A,\Omega,T,O,R,\gamma,b_0)$,其中,$S,A$ 和Ω分别表示有限且离散的状态空间、行动空间和观察空间,$T(s,a,s'):S\times A\times S\to[0,1]$为状态转移函数$(\Pr(s'|s,a))$,$O(a,s',o):A\times S\times\Omega\to[0,1]$是观察函数$(\Pr(o|a,s'))$,$R(s,a):S\times A\to\mathbb{R}$为立即回报函数,$\gamma\in(0,1)$是折扣因子,$b_0$为主体的初始信念状态.因为主体的当前状态是不完全可观察的,所以主体需要依赖过去行动和观察的完整历史信息来判断当前应该采取的行动.信念状态 b 是这些历史信息的充分统计量,它对应于状态空间上的一个离散的概率分布,其中的元素 $b(s)$表示主体的状态为 s 的概率,且满足 $\sum_{s\in S}b(s)=1$. B 表示所有可能信念状态构成的空间.当主体在信念状态 b 采取行动 a 得到观察 o 后,它将到达一个新的信念状态 $b^{a,o}$:

$$b^{a,o}(s')=\frac{O(a,s',o)\sum_{s\in S}T(s,a,s')b(s)}{\sum_{o\in\Omega}O(a,s',o)\sum_{s\in S}T(s,a,s')b(s)} \tag{1}$$

其中,分母表示的是在 b 处执行 a 得到 o 的概率 $\Pr(o|b,a)$.

最优值函数 V^*由从任意信念状态 $b\in B$ 开始通过执行最优策略π^*得到的最大期望折扣回报构成.它可以由著名的 Bellman 方程 $V^*(b)=\max_{a\in A}Q^*(b,a)$[16]计算得到,其中,

$$Q^*(b,a)=R(b,a)+\gamma\sum_{o\in\Omega}\Pr(o\mid b,a)V^*(b^{a,o}),R(b,a)=\sum_{s\in S}R(s,a)b(s).$$

本文把 $V^*(b)$称为 b 处的最优值,把 $Q^*(b,a)$称为(b,a)处的最优值.任意 POMDP 问题的最优值函数 V^*可以由分段线性凸函数 $V(b)=\max_{\alpha\in\Gamma}(\alpha\cdot b)=\max_{\alpha\in\Gamma}\sum_{s\in S}\alpha(s)b(s)$ 无限逼近,其中,Γ是一组α向量的有限集合.它的这个性质已经被许多精确和近似规划算法用在了其求解 POMDP 规划问题的设计思想中[1,4].

1.2 近似在线规划算法

在线的 POMDP 算法大致可以分成 3 类:分支限界裁剪法、蒙特卡罗采样法和启发式搜索法[6].在本文中,我们关注其中的一类:启发式在线搜索算法.它的思想是:同时保持最优值函数的下界 $\underline{V}(b)$ 和上界 $\overline{V}(b)$, 并通过扩展一棵由从 b_c 开始采取某些策略可到达的信念状态构成的与或树,达到改进当前信念状态 b_c 处最优值的上下界的目的.这棵树中的每个或节点表示一个信念状态(节点).本文既用 b 表示一个或节点,又用它表示或节点对应的信念状态.如果树是自上向下延伸的,那么与节点(b,a),又称为 Q 节点,表示的是其上方信念节点 b 对应的行动选择 a.我们用 $\underline{Q}(b,a)$ 和 $\overline{Q}(b,a)$ 分别表示(b,a)处最优值 $Q^*(b,a)$的下界和上界,并定义 b 处的当前最好下界保证的行动为 $\pi_{best}(b)=\arg\max_{a'\in A}\underline{Q}(b,a')$. 如果 a 满足 $V^*(b)-Q^*(b,a)<\varepsilon$,那么我们称 a 为 b 处的ε-最优行动.

算法 1. 通用的在线 POMDP 求解器.

Function *OnlinePOMDPSolver*(b_0,τ,ε)

1: 初始化最优值函数的上下界;
2: $b_c=b_0$;
3: 构建一棵以 b_c 为根且只包含 b_c 的与或树;
4: **while** b_c 不是目标状态 **do**
5: $a=Search(b_c,\tau,\varepsilon)$;
6: 执行 a 并且获得一个新的观察 o;
7: $b_c=b_c^{a,o}$;
8: 更新与或树使得 b_c 成为新的根节点;
9: **end while**

算法 2. 策略搜索.

Function *Search*(b_c,τ,ε)

1: *StartTimer*();
2: **while** *ElapsedTime*()$\leqslant\tau$ and $\overline{V}(b_c)-\underline{V}(b_c)>\varepsilon$ **do**
3: $b^*=$*ChooseBestNodetoExpand*();
4: *Expand*(b^*);
5: *UpdateAncestors*(b^*);
6: **end while**
7: return $\arg\max_{a'\in A}\underline{Q}(b_c,a')$;

算法 1 是一个通用的在线 POMDP 求解器.它接受 3 个参数:b_0,τ和ε,其中,b_0 为初始的信念状态,τ为每一个策略搜索步所允许的计算时间的上界,ε是想得到的 b_c 处最优值上下界之差的精度.在线规划算法使用这棵树在策略搜索步和策略执行步间迭代.在所有的在线规划算法中,策略执行步(见算法 1 的第 6 行~第 8 行)是相同的.算法 2 是一个策略搜索函数.子函数 *ChooseBestNodetoExpand*()使用某种启发式搜索策略在叶子信念节点中找到下一个被扩展的节点,搜索策略的好坏决定了启发式在线搜索算法的性能.子函数 *Expand*()通过扩展一个叶子信念节点来改进该节点处的最优值的上下界.子函数 *UpdateAncestors*()用于更新一个信念节点的所有祖先与

节点和祖先或节点处最优值的上下界.算法 2 中第 7 行返回的是 b_c 处的当前最好下界保证的行动.

我们用图 1 来进一步解释启发式在线搜索算法的工作原理.假设 b_{c+k} 是子函数 *ChooseBestNodetoExpand*() 返回的叶子信念节点.令 b_{c+k} 为从 b_c 采取 k 步行动-观察序列 $a_0o_1a_1o_2\ldots a_{k-1}o_k$ 到达的叶子信念节点.扩展 b_{c+k} 后,可以得到改进的 $\underline{V}(b_{c+k})$ 和 $\bar{V}(b_{c+k})$.进一步地,调用子函数 *UpdateAncestors*()将使得这条路径上的所有信念节点和 Q 节点处有改进的最优值的上下界.换句话说,如果 b_{c+k} 是通过采取某个策略 π 从 b_c 可达的,即 $\Pr(a_0o_1a_1o_2\ldots a_{k-1}o_k|b_c,\pi)>0$,那么算法 2 中第 3 行~第 5 行的执行将改进从 b_c 处起执行 π 得到的期望折扣回报的下界 $\underline{V}^{\pi}(b_c)$ 和上界 $\bar{V}^{\pi}(b_c)$.如果在调用 *UpdateAncestors*()之前的 $\underline{V}^{\pi_{best}}(b_c)$ 小于改进的 $\underline{V}^{\pi}(b_c)$,那么算法就找到了一个有更高下界保证的策略.

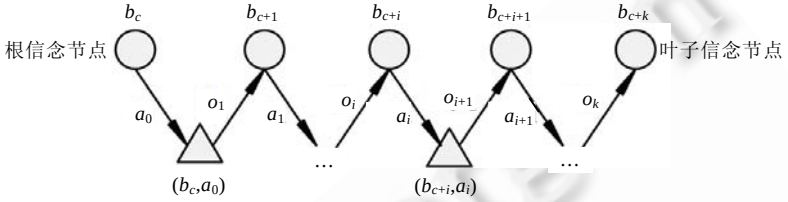

Fig.1 A path from the root belief node b_c to the leaf belief node b_{c+k}

图 1 从根信念节点 b_c 到叶子信念节点 b_{c+k} 的一条路径

1.3 现有在线方法里采用的启发式搜索函数

在现有的在线搜索算法中,每一个叶子信念节点都有一个启发值.子函数 *ChooseBestNodetoExpand*()返回的是具有最大启发值的叶子信念节点.如果最优值函数 V^* 已知,那么我们就可以通过如下公式来定义一个启发式搜索函数,它具有很好的理论保证[6]:

$$H^*(b_{c+k})=e^*(b_{c+k})\prod_{t=0}^{k-1}\omega^*(b_{c+t},a_t)\omega(b_{c+t},a_t,o_{t+1}) \tag{2}$$

其中, $e^*(b_{c+k})=V^*(b_{c+k})-\underline{V}(b_{c+k})$, $\omega(b_{c+t},a_t,o_{t+1})=\gamma\Pr(o_{t+1}\mid b_{c+t},a_t)$,

$$\omega^*(b_{c+t},a_t)=\begin{cases}1, \text{ if } a_t\in\arg\max_{a'\in A}Q^*(b_{c+t},a')\\ 0, \text{ otherwise}\end{cases} \tag{3}$$

在这个启发式函数里,每一项都在选择下一个被扩展的叶子信念节点中发挥着作用:$e^*(b_{c+k})$鼓励扩展最优值与下界之差较大的叶子信念节点,$\omega^*(b_{c+t},a_t)$把搜索的范围缩小到那些从 b_c 处起采取最优策略可达的叶子信念节点,$\omega(b_{c+t},a_t,o_{t+1})$把搜索导向将来最有可能遇到的信念节点.以往文献的实验结果告诉我们:忽略公式(2)中的一些项将会损害在线规划算法的整体性能.例如:BI-POMDP 算法[17]省去了 $\omega(b_{c+t},a_t,o_{t+1})$这一项,从而导致它在求解一些具有大规模观察空间的 POMDP 规划问题时,如将要提到的 FVRS_5_7 问题[15],性能较差,详细数据请参见文献[6]中的表 4.

然而,构造 H^* 所需的最优值函数或最优策略是不可获得的.于是,如何找到一组有前景的策略就成为摆在改进在线规划算法性能面前的一个核心问题.Satia&Lave 算法[18]假设这组有前景的策略是由所有可能的最优策略构成的.它很少利用上下界把搜索导向看起来有前景的行动分支.因此,Satia&Lave 算法不能很好地解决大规模的 POMDP 规划问题.AEMS1 算法[15]同时使用 Q 节点处最优值的上下界把搜索导向那些 $[\underline{Q}(b,a)+\bar{Q}(b,a)]/2$ 大的行动分支.该算法在处理大规模 POMDP 规划问题时会不高效,这归咎于它没有很好地利用最高的下界或最高的上界.与 AEMS1 算法不同,BI-POMDP 算法和 AEMS2 算法使用 $\bar{V}$ 作为 V^* 的近似替代.公式(4)中的 $H_U(b_{c+k})$为 AEMS2 算法中使用的基于上界的启发式搜索函数:

$$H_U(b_{c+k})=e(b_{c+k})\prod_{t=0}^{k-1}\omega(b_{c+t},a_t)\omega(b_{c+t},a_t,o_{t+1}) \tag{4}$$

其中, $e(b_{c+k})=\bar{V}(b_{c+k})-\underline{V}(b_{c+k})$,

$$\omega(b_{c+t},a_t)=\begin{cases}1, \text{ if } a_t\in\arg\max_{a'\in A}\bar{Q}(b_{c+t},a')\\ 0, \text{ otherwise}\end{cases} \tag{5}$$

由$\omega(b_{c+t},a_t)$的定义可见,AEMS2 算法假设具有最高上界的行动为最优行动.由这种假定构成的启发式方法被称为 IE-MAX 启发法[3].

2 杂合启发式在线规划算法(HHOP)

近年来,有一些文献研究了如何在 POMDP 在线规划算法中定义高效的启发式搜索函数,以及能保证算法收敛性的启发式搜索函数的特点[6,15,19].然而,在现有的启发式方法中,依然存在着一些尚未被利用的、重要的启发式信息,我们可以利用这些信息来进一步提高现有启发式方法的性能.在这一节里,我们首先构造一个基于下界的启发式搜索函数,它可以弥补 AEMS2 算法的启发式函数的不足;然后,我们通过一种杂合的方式把该基于下界的启发式搜索函数与 AEMS2 算法中的启发式搜索函数相结合,达到优势互补和避免各自不足的目的.这一做法的贡献在于提出了一种新的算法设计思路,即适当地使用下界作为启发式信息,能够带来一种比 AEMS2 更高效的启发式搜索算法.

2.1 使用下界来构造一个启发式搜索函数

AEMS2 算法中使用的启发式搜索函数能够保证算法在有限时间内找到 b_c 处的ε-最优行动.在这种算法中,沿着最高下界对应的行动分支进行搜索的想法被有意地避免,理由是扩展由b_c开始通过π_{best}可达的叶子信念节点只会使改进的$\underline{V}^{\pi_{best}}(b_c)$更大,从而使算法无法发现具有更好下界保证的策略,也就是说,会把搜索引向到局部最优.然而,这个理由是不充分的,因为我们没有必要让从 b_c 到叶子信念节点这条路径上的每个行动选择都遵从π_{best}.如果有一个行动选择不遵从π_{best},那么我们不仅可以用最高的下界作为启发式信息来找下一个被扩展的叶子信念节点,而且找到的节点并不是从 b_c 通过π_{best} 可达的.本节将利用这一想法来构造一个基于下界的启发式搜索函数.

本文第 1.2 节已经提到,为了找到一个更好的策略,我们需要选择一个策略π,然后扩展从 b_c 根据策略π可达的叶子信念节点.我们的目标是,使用下界作为启发信息来帮助算法尽快发现更好的策略.我们的主要想法是:首先,根据下界从可能的策略中提取出一组有前景的策略,这组策略中的任意一个策略π都应满足$\underline{V}^{\pi}(b_c)$接近于$\underline{V}^{\pi_{best}}(b_c)$;然后,扩展某些通过这组策略可达的叶子信念节点,达到发现更好的策略的目的.

首先,定义$\omega_1(b,a)$如下:

$$\omega_1(b_{c+t},a_t)=\begin{cases}1, \text{ if } a_t\in\arg\max_{a'\in A}\underline{Q}(b_{c+t},a')\\ 0, \text{ otherwise}\end{cases} \tag{6}$$

它能使搜索朝向那些从 b_c 采取当前最好下界保证的策略π_{best}可达的信念节点.

其次,考虑信念节点 b 处的行动分支.假设 b 处的当前最好下界保证的行动(集合)A_L为 $\arg\max_{a'\in A}\underline{Q}(b,a')$, 并定义 $A_S=\{a\in A\setminus A_L\mid \bar{Q}(b,a)>\max_{a'\in A}\underline{Q}(b,a')\}$.这样,$A_S$ 为所有行动中排除了 A_L 和次优行动分支后得到的行动集合.其中, $\bar{Q}(b,a)>\max_{a'\in A}\underline{Q}(b,a')$ 被用来裁剪掉 b 处所有的次优行动分支,这与 Satia&Lave 算法中的分支限界裁剪技术是类似的.

接下来,定义$\omega_2(b,a)$如下:

$$\omega_2(b_{c+t},a_t)=\begin{cases}1, \text{ if } a_t\in\arg\max_{a'\in A_S}\underline{Q}(b_{c+t},a')\\ 0, \text{ otherwise}\end{cases} \tag{7}$$

其中, $\arg\max_{a'\in A_S}\underline{Q}(b_{c+t},a')$ 被称为 b 处的当前第二好下界保证的行动(集合).$\omega_2(b,a)$把搜索导向那些从 b 处的当前第二好下界保证的行动分支可达的信念节点.我们使用图 2 给这些概念提供一个直观的解释.在该例中,$A_L=\{a_2\}$,$A_S=\{a_1,a_4\}$,当前第二好下界保证的行动为 a_4.a_3 没有被包含在 A_S 中是因为 $\bar{Q}(b,a_3)<\underline{Q}(b,a_2)$.

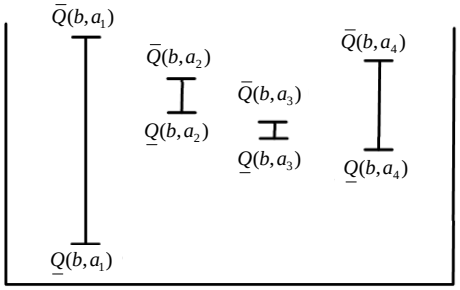

Fig.2 An example of action branches at b

图 2 一个信念状态 b 处行动分支的例子

最后,我们描述一种根据下界提取一组有前景的策略的方法.每一个有前景的策略对应一条从 b_c 开始到达叶子信念节点的特殊路径.其特殊之处在于:在所经过的所有非叶子信念节点中,有且仅有一个信念节点选择了当前第二好下界保证的行动分支,而所有其他的信念节点处都选择了当前最好下界保证的行动分支.这样一种策略可以通过$\omega_{12}(b_{c+k})$来加以描述:

$$\omega_{12}(b_{c+k})=\max_{i\in\{0,1,\ldots,k-1\}}\omega_2(b_{c+i},a_i)\prod_{\substack{t=0\\t\neq i}}^{k-1}\omega_1(b_{c+t},a_t) \tag{8}$$

其中,b_{c+i}即为选择了导向第二好下界保证的行动分支的节点.基于公式(8),我们定义“有前景的策略集合”:

$$\Pi_{promising}=\{\pi\in\Pi_{\text{POMDP}}|\Pr(b_{c+k}|b_c,\pi)>0,\omega_{12}(b_{c+k})=1\},$$

其中,Π_{POMDP}是指由生成当前与或树所需的策略构成的集合.这样,当$\omega_{12}(b_{c+k})=1$时,b_{c+k}肯定是从 b_c开始可以通过$\Pi_{promising}$中的一个到达的叶子信念节点.由公式(8)易得:当所有叶子信念节点均有 $k=N$ 时,$\Pi_{promising}$的个数为N.我们给每个叶子信念节点构造一个新的启发式函数 $H_L(b)$:

$$H_L(b_{c+k})=e(b_{c+k})\omega_{12}(b_{c+k})\prod_{t=0}^{k-1}\omega(b_t,a_t,o_{t+1}) \tag{9}$$

由公式(9)可知,使得启发值$H_L(b_{c+k})$大于 0 的叶子信念节点 b_{c+k}并不是可以从 b_c开始通过采取当前最好下界保证的策略到达的.然而,它们可以从 b_c开始通过采取有前景的策略集合$\Pi_{promising}$中的一个到达,$\Pi_{promising}$中的每一个策略与当前最好下界保证的策略都非常相似,因为它们仅仅在一个行动分支上做出了不同的选择.这样一个启发式搜索函数 $H_L(b)$具有以下优点:

(1) 具有更好下界保证的策略能够被快速发现.这是因为,使用 $H_L(b_{c+k})$作为启发式函数会引导算法去更新与当前最好下界保证的策略π_{best}相似的策略π的期望折扣回报的下界$\underline{V}^{\pi}(b_c)$.在更新之前,$\underline{V}^{\pi}(b_c)$略小于$\underline{V}^{\pi_{best}}(b_c)$;在更新之后,$\underline{V}^{\pi}(b_c)$会有所改进.如果改进后的$\underline{V}^{\pi}(b_c)$大于$\underline{V}^{\pi_{best}}(b_c)$,那么$\pi$就是新的具有最好下界保证的策略;

(2) 能够把搜索导向有前景的策略集合$\Pi_{promising}$而不是单一的策略,同时能够选择通过$\Pi_{promising}$中最具潜力的策略可达的叶子信念节点进行扩展.总是扩展单一策略可达的叶子信念节点的主要弊端在于:如果该策略不比当前最好下界保证的策略更好,那么扩展这些节点的时间就被浪费.由于 $H_L(b)$选取的是一组有前景的策略,当中次优的策略将随着待扩展的叶子信念节点深度的增加而逐渐被淘汰;

(3) 这组有前景的策略仅仅是所有可能的策略Π_{POMDP}中非常小的一部分.例如,当所有叶子节点均有$k=N$时,策略集合Π_{POMDP} 中策略的个数为$|A|^N$,这样,$\Pi_{promising}$ 中的策略个数仅占Π_{POMDP} 中策略个数的$N/|A|^N$.因此,这种技术能够避免搜索大规模的策略空间,满足在线规划算法实时性方面的要求.

然而,当最优策略并不在这个有前景的策略集和当前最好下界保证的策略中时,这种启发式方法仍可能使搜索陷入到局部最优.因而,新的启发式函数 $H_L(b)$可能只提供一种贪心策略,有时并不能把搜索导向全局最优.

2.2 构造一个杂合的启发式策略

在构建一个杂合策略之前,我们回顾一下 $H_L(b)$和 $H_U(b)$的优、缺点.$H_L(b)$是一个依赖于最优值函数下界的启发式搜索函数.其长处是可尽快地使搜索朝向那些有着更高下界保证的策略.然而,$H_L(b)$仅把搜索限制于从 b_c 通过一组有前景的策略可达的信念节点中.如果当前最好下界保证的策略和这组有前景的策略都是次优的,那么 $H_L(b)$可能会把搜索引向局部最优.$H_U(b)$是一个基于上界的启发式策略.如果时间足够充裕,那么 $H_U(b)$最终能把搜索导向到 b_c 处的ε-最优行动.然而,在处理有大规模行动空间和观察空间的 POMDP 规划问题时,这个寻找ε-最优行动的过程通常很缓慢.正因如此,$H_L(b)$和 $H_U(b)$的优点可以互补,把它们结合起来获得一个杂合启发式搜索策略是一个值得尝试的想法.

我们提出如下一种构建杂合策略的方法.令 LS 是与或树中的所有叶子信念节点构成的集合:

$$b_U=\max_{b\in LS}H_U(b) \tag{10}$$

$$b_L=\arg\max_{b\in LS}H_L(b) \tag{11}$$

那么,我们的杂合策略根据公式(12)选择出 b^*:

$$b^*=\begin{cases}b_U, \text{ if } C_UH_U(b_U)>C_LH_L(b_L)\\ b_L, \text{ otherwise}\end{cases} \tag{12}$$

其中,C_i 表示选择 b_i 来扩展能给 $\underline{V}(b_c)$ 和 $\overline{V}(b_c)$ 带来的期望改进值.我们在实验中使用公式(13)来计算 C_i:

$$C_i=\frac{I_i+1}{N_i+1} \tag{13}$$

其中,$i=U$ 或 L,N_i 表示到目前为止已扩展 b_i 的次数.I_i 表示由扩展 N_i 次 b_i 所带来的 $\underline{V}(b_c)$ 和 $\overline{V}(b_c)$ 的改进值的总和.在每次调用 $Search(b_c,\tau,\varepsilon)$时,$N_i$ 和 C_i 都会先被重置为 0.这里,b_U 表示的是所有由 $C_UH_U(b_U)>C_LH_L(b_L)$得到的 b^*,b_L 表示的是所有由 $C_UH_U(b_U)\leqslant C_LH_L(b_L)$得到的 b^*.分子和分母处出现的 1 用于表示在调用函数 $Search(b_c,\tau,\varepsilon)$时,$C_U$ 和 C_L 均被初始化为 1,从而避免公式(13)中出现 0/0 的情况.在新的杂合启发式搜索中,C_U 和 C_L 分别被用来调整 $H_U(b)$和 $H_L(b)$的权重.例如,如果在一段时间内扩展 b_U 仅仅带来 $\underline{V}(b_c)$ 和 $\overline{V}(b_c)$ 很小的改进,那么随着时间的推移,C_U 的值将减小,选择 b_U 来扩展的概率也将逐渐降低.

我们把这种基于杂合启发式策略的在线规划算法称为 HHOP.值得注意的是,我们的启发式策略与 AEMS1 算法中的策略有着本质的区别.AEMS1 算法在设计启发式搜索函数的行动选择策略时,把最优值函数的下界和上界合并在了一起.然而,HHOP 算法首先把最优值函数的下界和上界分隔开,以便利用它们来构造两个独立的启发式搜索函数;然后,HHOP 算法用一种杂合的方式把这两个函数结合在一起,达到了充分利用各自长处的目的.下面的性质 1 说明,HHOP 算法与 AEMS2 算法一样,也能够满足收敛性的要求.

性质 1. 令$\varepsilon>0$,b_c 为当前信念状态,则 HHOP 算法能够保证在有限时间内找到 b_c 处的ε-最优行动.

证明:根据文献[19]中的定理 2,我们只需证明:如果没有找到 b_c 处的ε-最优行动,那么 HHOP 算法将选择 b_U 扩展的概率要大于 0.下面用反证法来证明这一点.

假设在第 M 次扩展 b_L 或 b_U 后,HHOP 算法将不再选择 b_U 进行扩展,即 $C_UH_U(b_U)\leqslant C_LH_L(b_L)$总成立.那么,$I_i$ 和 N_i 将不再会改变,因此,C_U 和 $H_U(b_U)$在第 M 次扩展后也不会改变.令 $\underline{V}_0(b)$ 和 $\overline{V}_0(b)$ 分别为 b 处最优值的初始上下界,$e_0(b)=\overline{V}_0(b)-\underline{V}_0(b)$,则可以得到 $H_L(b_L)\leqslant e(b_L)\leqslant\max_{b\in B}e_0(b)$和 $I_L\leqslant e_0(b_c)$.

现令 $N_L>\max\left\{M,\dfrac{[e_0(b_c)+1]\max_{b\in B}e_0(b)}{C_UH_U(b_U)}-1\right\}$,则有

$$C_UH_U(b_U)>\frac{[e_0(b_c)+1]\max_{b\in B}e_0(b)}{N_L+1}\geqslant\frac{I_L+1}{N_L+1}H_L(b_L)=C_LH_L(b_L).$$

这与之前的假设相矛盾.证毕. □

值得一提的是,还有很多结合基于下界和基于上界的启发式函数的方法.在这里,我们仅描述了一种可行的杂合方法.对这个问题的进一步研究可能会产生更高效的杂合方法.

3 实验结果

在这一节里,我们首先描述本文的实验平台,包括所测问题的参数;然后,列出详细的实验结果并进行分析,这些结果反映出我们的新算法的整体性能;最后,通过一组辅助的实验数据,进一步解释新算法更高效的内在原因.

3.1 实验平台

在所有的实验中,我们使用的是盲目的策略(blind policy)方法[20]来初始化下界,并用快速通知界(fast informed bound,简称 FIB)方法[20]来初始化上界.除非特别说明,所有的实验都是在一台配置为 AMD 3600+ 2.00GHz 双核处理器和 2GB 内存的机器上完成的.我们使用 C++,基于一个现有的、高效的 POMDP 求解器 APPL-0.93[21]实现了 HHOP 算法和 AEMS2 算法,一种最近提出的被称为 Mixed Observability MDP(简称 MOMDP)的因子化状态表示方法[22]被使用在了这两种算法的实现过程中.因此,AEMS2 算法在这里的性能要比其在文献[6]中的性能好很多.我们在 9 个基准问题上比较了 HHOP 算法和一些现有算法的性能.表 1 列出了与这些问题相关的参数.这些问题的详细描述可参考文献[3−5,23,24].

Table 1 Benchmark problem parameters

表 1 基准问题的参数

Problem	$\|S\|$	$\|A\|$	$\|\Omega\|$	Problem	$\|S\|$	$\|A\|$	$\|\Omega\|$	Problem	$\|S\|$	$\|A\|$	$\|\Omega\|$
Hallway	61	5	21	RS_7_8	12 545	13	2	FVRS_5_5	801	5	32
Hallway2	93	5	17	RS_10_10	102 401	15	2	FVRS_5_7	3 201	5	128
Tag	870	5	30	RS_11_11	247 809	16	2	AUV navigation	13 536	6	144

注:RS=RockSample, FVRS=FieldVisionRockSample.

3.2 HHOP的实时性能及与相关算法的比较

我们利用新提出的 HHOP 算法和现有的一些在线与离线规划算法分别求解表 1 所列 9 个基准问题.表 2 列出了这些问题的求解结果.我们用 7 个度量[6]来评价这些算法在这些问题上的性能:期望折扣回报(reward)、每一个策略搜索步所允许的在线规划时间的上界(τ)、用于初始化最优值函数上下界的离线时间(offline time)、平均每个策略搜索步带来的 b_c 处的误差界 $e(b_c)$减小百分比(EBR)、平均每个策略搜索步带来的 b_c 处的最优值下界改进量(LBI)、以 b_c 为根节点的与或树中的信念节点的平均个数(belief nodes)、平均下一个策略搜索步中可重利用的信念节点数与当前与或树中总信念节点数的百分比(node reused).其中,前 3 个是主要的度量,后 4 个是辅助性的度量.EBR 可以公式化为 $1-[\bar{V}(b_c)-\underline{V}(b_c)]/[\bar{V}_0(b_c)-\underline{V}_0(b_c)]$,LBI 则为 $\underline{V}(b_c)-\underline{V}_0(b_c)$, 其中,$\bar{V}_0(b_c)$ 和 $\underline{V}_0(b_c)$ 分别为 b_c 处的最优值的初始上下界.有关这些辅助性度量在方法论上的意义,文献[6]中有详细的阐述.对于期望折扣回报和所有的辅助性度量,表 2 给出了它们的均值和对应的 95%的置信区间.为了保证这些度量的 95%的置信区间足够小,我们的算法对每个基准问题都进行了少则数百次,多则几千次的仿真实验.一般来讲,一种在线规划算法越高效,其对应的期望折扣回报值和 4 个辅助的度量值就越高.因为 AEMS2 算法的实时性能基本上要优于前面提到的 Satia&Lave,BI-POMDP 和 AEMS1 等算法[6],所以表 2 中仅包含 AEMS2 算法的实验结果.

在给定相同的在线规划时间上界τ的前提下,HHOP 算法在求解所有基准问题时所得到的策略的期望折扣回报均要高于 AEMS2 算法的相应回报,如在求解 FVRS_5_7 问题时,HHOP 算法的期望折扣回报 24.46 要高于 AEMS2 算法的期望折扣回报 23.13.4 个辅助性度量中的数据为分析 AEMS2 算法和 HHOP 算法的性能提供了详细的依据.这些结果为证明我们的杂合启发式策略比 AEMS2 算法中的启发式策略更高效提供了有力的实验证据.

表 2 中还列出了 SARSOP 算法的实验结果,其中,Hallway 类和 FVRS 类问题的结果来自于我们的实验平台,其他结果来自文献[5].通过比较可以发现,HHOP 算法非常具有竞争力.在求解许多问题时,HHOP 算法只需要少量的初始化规划时间和很少的重规划时间,却可以得到与 SARSOP 算法相媲美的策略.据我们所知,虽然还存在

其他的离线规划算法[3,5,25−28],但 SARSOP 是其中最高效的 POMDP 离线规划算法之一.这里,我们仅以 SARSOP 为代表与在线规划算法进行比较.

Table 2 Different algorithms' performance comparison on several standard benchmark problems

表 2 不同算法在多个基准问题上的性能比较

Method	Reward	τ (s)	Offline time (s)	EBR (%)	LBI	Belief nodes	Nodes reused (%)
Hallway							
AEMS2	0.50±0.01	0.20	0.02	18.3±0.2	0.08±0.01	1 651±15	11.9±0.2
HHOP	**0.52±0.01**	**0.20**	**0.02**	**29.0±0.2**	**0.14±0.01**	**3 110±24**	**20.0±0.2**
SARSOP	0.52±0.01	0.00	1.05	n.a.	n.a.	n.a.	n.a.
Hallway2							
AEMS2	0.33±0.01	0.20	0.03	14.8±0.1	0.06±0.01	1 198±14	20.1±0.3
HHOP	**0.36±0.01**	**0.20**	**0.03**	**17.0±0.1**	**0.08±0.01**	**1 718±20**	**27.2±0.3**
SARSOP	0.35±0.01	0.00	1.16	n.a.	n.a.	n.a.	n.a.
Tag							
AEMS2	−6.14±0.14	0.10	0.12	81.2±0.3	13.37±0.05	90 931±149	56.2±0.2
HHOP	**−6.04±0.14**	**0.10**	**0.12**	**85.9±0.3**	**14.47±0.05**	**119 116±158**	**67.8±0.2**
SARSOP[5]	−6.03±0.12	0.00	16.50	n.a.	n.a.	n.a.	n.a.
RS_7_8							
AEMS2	21.11±0.29	0.10	0.51	59.7±0.2	5.57±0.03	4 068±17	43.2±0.3
HHOP	**21.45±0.30**	**0.10**	**0.51**	**63.1±0.2**	**5.92±0.04**	**5 833±20**	**51.1±0.4**
POMCP[12]	20.71±0.21	1.00	n.v.	n.v.	n.v.	n.v.	n.v.
SARSOP[5]	21.39±0.01	0.00	810.00	n.a.	n.a.	n.a.	n.a.
RS_10_10							
AEMS2	20.90±0.28	1.00	2.82	58.6±0.1	5.65±0.03	6 925±51	43.7±0.2
HHOP	**21.35±0.34**	**1.00**	**2.82**	**63.4±0.2**	**5.92±0.03**	**8 832±58**	**50.2±0.2**
SARSOP[5]	21.47±0.11	0.00	1 589.00	n.a.	n.a.	n.a.	n.a.
RS_11_11							
AEMS2	21.29±0.29	1.00	7.97	43.1±0.1	5.38±0.03	6 084±41	37.2±0.2
HHOP	**21.49±0.32**	**1.00**	**7.97**	**47.2±0.1**	**5.57±0.03**	**7 734±43**	**45.6±0.2**
POMCP[12]	20.01±0.23	1.00	n.v.	n.v.	n.v.	n.v.	n.v.
SARSOP[5]	21.56±0.11	0.00	1 369.00	n.a.	n.a.	n.a.	n.a.
FVRS_5_5							
AEMS2	21.33±0.34	0.20	0.02	34.1±0.2	3.27±0.03	14 308±476	4.1±0.1
HHOP	**22.56±0.33**	**0.20**	**0.02**	**36.2±0.2**	**3.48±0.03**	**18 190±512**	**4.5±0.1**
SARSOP	23.20±0.33	0.00	508.40	n.a.	n.a.	n.a.	n.a.
FVRS_5_7							
AEMS2	23.13±0.33	0.20	0.12	15.2±0.3	2.88±0.06	2 657±59	2.6±0.1
HHOP	**24.46±0.34**	**0.20**	**0.12**	**16.4±0.3**	**3.11±0.06**	**4 217±64**	**3.8±0.1**
SARSOP	29.48±0.34	0.00	1 029.38	n.a.	n.a.	n.a.	n.a.
AUV Navigation							
AEMS2	1 047.29±8.41	10.00	16.02	90.8±0.1	1 818.95±4.38	171 910±386	74.4±0.2
HHOP	**1 059.22±8.49**	**10.00**	**16.02**	**92.2±0.1**	**1 867.45±4.45**	**195 692±393**	**78.4±0.2**
SARSOP[5]	1 019.80±9.70	0.00	409.00	n.a.	n.a.	n.a.	n.a.

注:n.a.=not applicable, n.v.=not available.

另外,文献[12]中有关部分可观察的蒙特卡罗规划(partially observable Monte-Carlo planning,简称 POMCP)算法在 RS 类问题中的一些结果也被放在了表 2 中.POMCP 是求解大规模 POMDP 规划问题的、另一类有前景的在线规划算法.从表 2 中的数据可以得出,无论从计算时间还是从计算得到的策略的质量来看,HHOP 算法都超过了 POMCP 算法.

3.3 辅助的实验结果

这里,我们不再进一步对 HHOP 算法在各基准问题上的辅助性度量值作详细分析.这些分析仅能为 HHOP 算法比 AEMS2 算法更高效提供更多的实验依据,而不能让我们清楚地看到 H_L 和 H_U 在 HHOP 算法的运行中扮演的角色.我们期望得到在 HHOP 算法的 *Search* 函数执行的某一时刻,用于计算公式(13)中变量 C_U 和 C_L 的变量 I_U,I_L,N_U 和 N_L 的值,它们能够告诉我们启发式搜索函数 H_L 和 H_U 分别带来的 b_0 处的上下界改进值和被扩展的叶子信念节点分别来自于 b_L 和 b_U 的数量.为了获得这组数据,我们设置 *Search* 函数中的参数分别为

$b_c=b_0$, τ=100s 和 ε=0.01.

表 3 中显示的是 *Search* 函数运行到 100s 时 I_U,I_L,N_U 和 N_L 的值.I_i 被写成了“扩展 b_i 带来的 b_c 处最优值下界的累积改进量+扩展 b_i 带来的 b_c 处最优值上界的累积改进量”的形式.例如,第 2 行第 2 列中的 0.02+0.04 中的 0.02 表示的是在 100s 内扩展 b_L 带来的 $\underline{V}(b_c)$ 的累积改进量.

Table 3 Statistics about hybrid heuristics on benchmark problems

表 3 杂合启发式法在各基准问题上的统计数据

Problem	I_U	N_U	I_L	N_L	Problem	I_U	N_U	I_L	N_L
Hallway	0.02+0.04	564	0.04+0.01	548	RS_11_11	4.53+002.71	9 082	4.02+00.15	4 569
Hallway2	0.01+0.04	494	0.04+0.00	410	FVRS_5_5	1.09+001.30	5 629	2.27+00.25	6 121
Tag	5.84+5.41	48 019	7.98+0.58	36 511	FVRS_5_7	1.82+001.55	515	2.95+00.19	441
RS_7_8	4.67+2.84	16 252	3.89+0.43	10 130	AUV	528.26+141.20	68 493	351.24+21.26	36 639
RS_10_10	7.65+2.42	9 174	4.07+0.38	3 587	Navigation				

从表 3 中我们可以看到,在所有测试的基准问题中,N_U 占了总数 N_U+N_L 的一半多.例如,Hallway 问题中有 N_U=564>N_L=548,这说明 H_U 在 b_c 处最优值上下界的改进中依然扮演了重要的角色.再进一步观察 I_U 的组成会发现,I_U 的值很大一部分来自于扩展 b_U 带来的 b_c 处最优值上界 $\bar{V}(b_c)$ 的减少量,而 I_L 的值更经验地依赖于 b_c 处最优值下界 $\underline{V}(b_c)$ 的改进值.特别地,对于大部分基准问题而言,扩展 b_L 带来的 $\underline{V}(b_c)$ 的改进值要大于扩展 b_U 带来的 $\underline{V}(b_c)$ 的改进值.基于这组数据,我们有理由推断如下过程经常发生在 HHOP 算法的运行中:当 H_U 带来的 b_c 处最优值上下界的改进速度变得缓慢时,算法会让 H_L 接管对下一步被扩展的叶子信念节点的控制权.H_L 通常会在此时提供更有前景的叶子信念节点进行扩展,从而有机会带来 $\underline{V}(b_c)$ 更大的改进.这样,在搜索的早期,与或树中许多次优的分支会由于它们对应策略的期望折扣回报值小于 $\underline{V}(b_c)$ 而被裁剪掉,这为 HHOP 算法比其他在线规划算法更高效提供了更深入的解释.

最后需要指出的是,第 2.2 节给出的杂合方式并不一定是最优或近似最优的.如何在基于下界的启发式搜索函数和基于上界的启发式搜索函数之间加以权衡是一个复杂的研究课题.我们还尝试了其他的杂合方法,最简单的方法是让 b_U 和 b_L 轮流地选择下一个被扩展的叶子信念节点.我们发现,这些杂合方法在求解某些基准问题时尤为高效,而在处理其他基准问题时就逊色一些.到目前为止,HHOP 算法中使用的杂合方法在所有测试问题上的综合性能是最好的.

4 结束语

本文介绍了一个新颖的杂合启发式搜索函数,其作用是加速现有的在线 POMDP 规划算法.在这个杂合函数的构造过程中,我们设计了一个基于最优值函数下界的启发式搜索函数,它的实现巧妙地利用了现有方法中没有被很好利用的、但仍很重要的下界信息.我们的杂合启发法提供了一种充分利用最优值函数上界和下界的方法.在实验部分,首先对新的在线规划算法 HHOP 和现有的在线与离线规划算法进行了系统的比较,然后利用观察到的统计数据对 HHOP 算法的高效性进行了深入的解释.在未来的工作中,我们将探索其他更为高效的杂合启发式搜索函数,以进一步提高现有在线规划算法的性能.

致谢 在此,我们向对本文的工作给予帮助和建议的罗格斯大学的 Michael L. Littman 教授、新加坡国立大学的 David Hsu 副教授、卡内基梅隆大学的 Stephane Ross 博士以及讨论班上的吴锋同学表示感谢.

References:

[1] Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. Artificial Intelligence (AIJ), 1998,101(1-2):99–134.

[2] Madani O, Hanks S, Condon A. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In: Proc. of the Association for Advancement of Artificial Intelligence (AAAI 1999). 1999. 541–548. http://www.informatik.uni-trier.de/~ley/db/conf/aaai/aaai99.html

[3] Smith T, Simmons R. Heuristic search value iteration for POMDPs. In: Proc. of the 20th Conf. on Uncertainty in Artificial Intelligence (UAI 2004). 2004. 520−527. http://www.informatik.uni-trier.de/~ley/db/conf/uai/uai2004.html

[4] Pineau J, Gordon G, Thrun S. Anytime point-based approximations for large POMDPs. Journal of Artificial Intelligence Research (JAIR), 2006,27:335−380. [doi: 10.1613/jair.2078]

[5] Kurniawati H, Hsu D, Lee WS. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: Proc. of the Robotics: Science and Systems IV (RSS 2008). 2008. http://www.roboticsproceedings.org/

[6] Ross S, Pineau J, Paquet S, Chaib-Draa B. Online planning algorithms for POMDPs. Journal of Artificial Intelligence Research (JAIR), 2008,32(1):663−704.

[7] Poupart P. Exploiting structure to efficiently solve large scale partially observable Markov decision processes POMDPs [Ph.D. Thesis]. University of Toronto, 2005.

[8] Zhang B, Cai QS, Guo BN. POMDP system and its solution for spoken dialogue system. Journal of Computer Research and Development, 2002,39(2):217−224 (in Chinese with English abstract).

[9] Li X, Chen XP. A real-time planning system in dynamic nondeterministic environments. Chinese Journal of Computers, 2005,28(7): 1163−1170 (in Chinese with English abstract).

[10] Hsiao K, Kaelbling LP, Lozano-Pérez T. Grasping POMDPs. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA 2007). 2007. 4485−4492. [doi: 10.1109/ROBOT.2007.364201]

[11] Qian K, Ma XD, Dai XZ, Fang F. POMDP navigation of service robots with human motion prediction. Robot, 2010,32(1):18−33 (in Chinese with English abstract). [doi: 10.3724/SP.J.1218.2010.00018]

[12] Silver D, Veness J. Monte-Carlo planning in large POMDPs. In: Proc. of the 24th Annual Conf. on Neural Information Processing Systems (NIPS 2010). 2010. 2164−2172. http://www.informatik.uni-trier.de/~ley/db/conf/nips/nips2010.html

[13] Hsu D, Lee WS, Rong N. What makes some POMDP problems easy to approximate? In: Proc. of the 21st Annual Conf. on Neural Information Processing Systems (NIPS 2007). 2007. 28−35. http://www.informatik.uni-trier.de/~ley/db/conf/nips/nips2007.html

[14] Zhang ZZ, Littman ML, Chen XP. Covering number as a complexity measure for POMDP planning and learning. In: Proc. of the 26th AAAI Conf. on Artificial Intelligence (AAAI 2012). 2012. 1853−1859. http://www.informatik.uni-trier.de/~ley/db/conf/aaai/aaai2012.html

[15] Ross S, Chaib-Draa B. AEMS: An anytime online search algorithm for approximate policy refinement in large POMDPs. In: Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2007). 2007. 2592−2598. http://www.informatik.uni-trier.de/~ley/db/conf/ijcai/ijcai2007.html

[16] Bellman R. Dynamic Programming. Princeton University Press, 1957.

[17] Washington R. BI-POMDP: Bounded, incremental partially observable Markov model planning. In: Proc. of the 4th Euro. Conf. on Planning. 1997. 440−451. [doi: 10.1007/3-540-63912-8_105]

[18] Satia JK, Lave RE. Markovian decision processes with probabilistic observation of state. Management Science, 1973,20(1):1−13. [doi: 10.1287/mnsc.20.1.1]

[19] Ross S, Pineau J, Chaib-Draa B. Theoretical analysis of heuristic search methods for online POMDPs. In: Proc. of the 21st Annual Conf. on Neural Information Processing Systems (NIPS 2008). 2008. 1233−1240. http://www.informatik.uni-trier.de/~ley/db/conf/nips/nips2008.html

[20] Hauskrecht M. Value-Function approximations for partially observable Markov decision processes. Journal of Artificial Intelligence Research (JAIR), 2000,13:33−94. http://arxiv.org/abs/1106.0234

[21] M^2AP's APPL software package repository page. http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/index.php?n=Main.Download

[22] Ong SCW, Png SW, Hsu D, Lee WS. Planning under uncertainty for robotic tasks with mixed observability. Int'l Journal of Robotics Research (IJRR), 2010,29(8):1053−1068.

[23] Littman ML, Cassandra AR, Kaelbling LP. Learning policies for partially observable environments: Scaling up. In: Proc. of the 12th Int'l Conf. on Machine Learning (ICML'95). 1995. 362−370. http://www.informatik.uni-trier.de/~ley/db/conf/icml/icml1995.html

[24] Pineau J, Gordon G, Thrun S. Point-Based value iteration: An anytime algorithm for POMDPs. In: Proc. of the 18th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2003). 2003. 1025−1030. http://www.informatik.uni-trier.de/~ley/db/conf/ijcai/ijcai2003.html

[25] Shani G, Brafman RI, Shimony SE. Forward search value iteration for POMDPs. In: Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2007). 2007. 2619−2624. http://www.informatik.uni-trier.de/~ley/db/conf/ijcai/ijcai2007.html

[26] Bian AH, Wang CJ, Chen SF. Preprocessing for point-based algorithms for POMDP. Ruan Jian Xue Bao/Journal of Software, 2008,19(6):1309−1316 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/1309.htm [doi: 10.3724/SP.J.1001.2008.01309]

[27] Bonet B, Geffner H. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In: Proc. of the 22nd Int'l Joint Conf. on Artificial Intelligence (IJCAI 2009). 2009. 1641−1646. http://www.informatik.uni-trier.de/~ley/db/conf/ijcai/ijcai2009.html

[28] Zhang Z, Chen X. Accelerating point-based POMDP algorithms via greedy strategies. In: Proc. of the Int'l Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2010). 2010. 545−556. http://www.informatik.uni-trier.de/~ley/db/conf/simpar/simpar2010.html

附中文参考文献:

[8] 张波,蔡庆生,郭百宁.口语对话系统的 POMDP 模型及求解.计算机研究与发展,2002,39(2):217−224.

[9] 李响,陈小平.一种动态不确定性环境下的持续规划系统.计算机学报,2005,28(7):1163−1170.

[11] 钱堃,马旭东,戴先中,房芳.预测行人运动的服务机器人 POMDP 导航.机器人,2010,32(1):18−33.

[26] 卞爱华,王崇骏,陈世福.基于点的 POMDP 算法的预处理方法.软件学报,2008,19(6):1309−1316. http://www.jos.org.cn/1000-9825/19/1309.htm [doi: 10.3724/SP.J.1001.2008.01309]

章宗长(1985－),男,江西修水人,博士生,主要研究领域为部分可观察的马氏决策过程,强化学习,预测状态表示,多主体合作及对抗.
E-mail: zzz@mail.ustc.edu.cn

陈小平(1955－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为人工智能逻辑,多主体系统,自主机器人系统关键技术.
E-mail: xpchen@ustc.edu.cn