一种基于最优策略概率分布的POMDP值迭代算法

刘 峰[1,3],王崇骏[2,3],骆 斌[1,3]
(1. 南京大学软件学院,江苏南京 210093;2. 南京大学计算机科学与技术系,江苏南京 210093;
3. 南京大学软件新技术国家重点实验室,江苏南京 210093)

摘 要: 随着应用中POMDP问题的规模不断扩大,基于最优策略可达区域的启发式方法成为了目前的研究热点.然而目前已有的算法虽然保证了全局最优,但选择最优动作还不够精确,影响了算法的效率.本文提出一种基于最优策略概率的值迭代方法PBVIOP.该方法在深度优先的启发式探索中,根据各个动作值函数在其上界和下界之间的分布,用蒙特卡罗法计算动作最优的概率,选择概率最大的动作作为最优探索策略.在4个基准问题上的实验结果表明PBVIOP算法能够收敛到全局最优解,并明显提高了收敛效率.

关键词: 部分可观测马尔科夫决策过程;基于最优策略概率的值迭代算法;蒙特卡罗法
中图分类号: TP319 **文献标识码:** A **文章编号:** 0372-2112 (2016)05-1078-07
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.05.010

A Probability-Based Value Iteration on Optimal Policy Algorithm for POMDP

LIU Feng[1,3], WANG Chong-jun[2,3], LUO Bin[1,3]
(1. *Software Institute, Nanjing University, Nanjing, Jiangsu* 210093, *China*;
2. *Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*;
3. *National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*)

Abstract: With the enlargement of the scale of POMDP problems in applications, the research of heuristic methods for reachable area based on the optimal policy becomes current hotspot. However, the standard of existing algorithms about choosing the best action is not perfect enough thus the efficiency of the algorithms is affected. This paper proposes a new value iteration method PBVIOP (Probability-based Value Iteration on Optimal Policy). In depth-first heuristic exploration, this method uses the Monte Carlo algorithm to calculate the probability of each optimal action according to the distribution of each action's Q function value between its upper and lower bounds, and chooses the maximum probability action. Experiment results of four benchmarks show that PBVIOP algorithm can obtain global optimal solution and significantly improve the convergence efficiency.

Key words: partially observable Markov decision process (POMDP); probability-based value iteration on optimal policy (PBVIOP); Monte Carlo method

1 引言

规划问题,即"设计合理的行动计划以达到个体目标"[1],是人工智能研究里的重要领域.序列决策问题(Sequential Decision Making)是规划问题的一个重要子领域.而动态不确定性环境下的行动规划是其中的热点,其动态性和不确定性是在这种环境下进行行动规划的主要难点.

部分可观察马氏决策过程(Partially Observable Markov Decision Process,POMDP)是一个强大的数学框架,可以用来描述并解决很多实际的不确定环境中序列决策问题,例如机器人探索任务[2]、口语对话管理[3]、服务漂移[4]、传感器调度[5]等.

精确求解POMDP问题计算复杂度过高,难以应用于实际问题,因此出现了各种近似算法如FIB[6]、MA-Q-learning[7]等等.其中基于点的值迭代方法在可达信念点集上进行迭代,通过增加迭代次数提升了整体效率,使得POMDP可以应用到较大规模的问题并在实际应用中取得了良好的效果.自从基于点的值迭代方法PBVI[8]提出之后,对探索信念点集的启发式探索方法成为了研究热点.PEMA[9]算法选取误差最大的后继点,使点迭代尽可能近似精确迭代;HSVI[10]、SARSOP[11]、GapMin[12]、PGVI[13]等算法根据最优值函数上界来选择最优动作探索最优可达信念点集,保证收敛到全局最

收稿日期:2014-09-15;修回日期:2015-03-19;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61375069);江苏省自然科学基金(No.BK20131277)

优;AEMS[14]、HHOP[15]等算法构造启发式函数选择最优动作探索最优可达信念点集,提高了收敛效率.

为了解决较大规模的 POMDP 问题,近年来基于点的算法通过探索最优可达信念空间来提高算法的效率.为了保证值函数能够收敛到全局最优解,HSVI 等算法在探索最优可达信念空间时,根据 IE-MAX[16]原则选取值函数上界最大的动作.但值函数的上界通过线性规划等方法来计算,其收敛效率很低,而值函数下界基于贝尔曼方程进行迭代收敛效率较高.HSVI 等算法虽然可以在理论上保证收敛,但在选择最优动作时仅以值函数上界为参照而完全不考虑值函数下界的取值情况,降低了值函数下界的迭代收敛效率,从而影响了算法的整体收敛效率.为保证高效地探索到全局最优解,HHOP 算法设计了有前景的策略再结合最优值函数上界构造了两个独立的启发式搜索函数进行杂合以探索最优可达信念空间.本文提出基于最优策略概率的值迭代算法(Probability-based Value Iteration on Optimal Policy,PBVIOP)来提高全局最优解的收敛效率.在探索最优可达信念空间时,PBVIOP 算法和 HHOP 算法一样都考虑了值函数的上界和下界,不同之处在于 HHOP 算法在每次探索时是把有前景的策略和值函数上界分隔开来各自考虑后再杂合;而 PBVIOP 算法在每次探索时先结合动作值函数的上界和下界来探索最优策略,再贪婪探索其不确定性最大的后继信念点,相比之下 HHOP 算法更为细致复杂.PBVIOP 算法在探索最优可达信念空间方面有如下特点:首先,在寻找最优策略的过程中同时参考动作值函数的上界和下界,保证算法的收敛质量和效率;其次,把选择最优动作建模成基于各个动作值函数的分布求最大值函数的问题,以各个动作值函数最大的概率作为选择最优动作的标准,保证了算法的可靠性和稳定性;最后,引入蒙特卡罗方法来近似计算动作最优的概率,使得算法合理且高效.算法在选择最优动作时避免了局部化的干扰,可以稳定达到全局最优.试验结果表明 PBVIOP 算法优于 HSVI 和 GapMin 算法的性能,且随着 POMDP 问题规模的扩大其优势愈加显著.

2　背景和相关工作

2.1　POMDP 模型

POMDP 模型可以表示为一个八元组(S,A,Z,b_0,T,O,R,γ)[8].其中 S 是一个隐含状态的有限集合,表示了系统所有可能处于的状态;A 是一个动作的有限集合,包括 Agent 能够采取的所有动作;Z 是一个观察的有限集合,表示 Agent 所有可能的输入;b_0是初始的状态分布,表示在初始时刻 t_0系统在状态集合 S 上的概率分布;$T(s,a,s')$是状态到状态的转移概率,描述 Agent 在状态 s 采取动作 a 后到达状态 s'的概率,表明了动作的随机效应;$O(a,s',z)$是 Agent 采取动作 a 到达状态 s'后且观察到 z 的概率,模拟了 Agent 部分可观测的特性;$R(s,a)$是在状态 s 时采取动作 a 所获得的回报值;$\gamma\in(0,1)\gamma\in(0,1)$是折扣因子.

在 POMDP 中,Agent 不能直接获取自己的状态而只能从环境中获得观察信息作为状态的参照,所以它必须根据动作和观测的历史序列$\{a_0,z_1,a_1,z_2,a_2,z_3,\cdots,a_{t-1},z_t\}$来决策下一个动作 a_t.因此 POMDP 引入维持历史信息的充分统计量 $\boldsymbol{b}$ 来代替历史序列以计算其长远回报[17].$\boldsymbol{b}$ 是一个代表状态上概率分布的向量:

$$b_t(s)=P(s_t=s\mid z_t,a_{t-1},\cdots,a_0)$$

在 POMDP 中 t 时刻的信念点 b_t可以根据贝叶斯规则来更新,只涉及前一步的信念状态 b_{t-1},最近采取的动作 a_{t-1}和得到的观测 z_t,因而 $\boldsymbol{b}$ 的更新具有 Markov 性.

$$b_t(s')=\tau(b_{t-1},a_{t-1},z_t)=\frac{O(s',a_{t-1},z_t)\sum_{s\in S}T(s,a_{t-1},s')b_{t-1}(s)}{P(z_t\mid b_{t-1},a_{t-1})}$$

$$P(z_t\mid b_{t-1},a_{t-1})=\sum_{s'\in S}O(s',a_{t-1},z_t)\sum_{s\in S}T(s,a_{t-1},s')b_{t-1}(s)$$

2.2　POMDP 求解

POMDP 中的策略是一个由信念到动作的映射:$\pi(\boldsymbol{b})\to a$. Agent 在策略 π 下的长远回报为:

$$V_\pi(\boldsymbol{b})=E\Big[\sum_{t=t_0}^{T}\gamma^{t-t_0}R(b_t,\pi(b_t))\Big]$$

POMDP 的求解是指 POMDP 模型完全已知(状态集合、动作集合、转移函数、回报函数等)的情况下计算最优策略π^*,它能够最大化长远回报的期望.最优策略可以由贝尔曼方程迭代获得.Q 值函数 $Q_{t+1}(b,a)$是 t 步视野内在当前信念点 $\boldsymbol{b}$ 处执行动作 a 的回报值:

$$Q_{t+1}(\boldsymbol{b},a)=\sum_{s\in S}b(s)R(s,a)+\gamma\sum_{z\in Z}P(z\mid\boldsymbol{b},a)\,V_t^*(\tau(\boldsymbol{b},a,z))$$

$$V_{t+1}^*(\boldsymbol{b})=\max_{a\in A}Q_{t+1}(\boldsymbol{b},a)$$

其对应的最优策略可以表示为:

$$\pi_{t+1}^*(\boldsymbol{b})=\arg\max_{a\in A}Q_{t+1}(\boldsymbol{b},a)$$

$\boldsymbol{b}$ 在$|S|-1$维的连续单空间△内有无限个取值,为了求解 POMDP 问题,Sondik 证明了对于任意的有限视野 t,值函数是信念空间上的分段线性凸函数[17],可表示为一个向量集合$\Gamma_t=\{\alpha_0,\alpha_1,\cdots\alpha_{|\Gamma_t|}\}$,$V_t(\boldsymbol{b})=\max_{\alpha\in\Gamma_t}\boldsymbol{b}\cdot\boldsymbol{\alpha}$.无限视野下的值函数可以被向量集合以任意小的误差逼近.由此可以获得精确求解 POMDP 问题的方法.首先获得一步回报集合:

$$\Gamma_1^a = \{\alpha^a \mid \alpha^a(s) = R(s,a)\ \forall s\}$$

$$\Gamma_1 = \bigcup_{a\in A}\Gamma_1^a$$

然后通过 update 操作由Γ_t获得Γ_{t+1}，update 分几个子步骤进行. 对于一个动作 a 和一个观察 z 来说，

$$\Gamma_{t+1}^{a,z} = \{\alpha_i^{a,z} \mid \alpha_i^{a,z}(s) = \gamma\sum_{s'\in S} T(s,a,s')O(s',a,z)\,\alpha_i'(s'),\ \alpha_i' \in \Gamma_t\}$$

再将这些集合与一步回报集合笛卡尔和相加得到某一动作 a 所对应的向量：

$$\Gamma_{t+1}^a = \Gamma_1^a \oplus \Gamma_{t+1}^{a,z_1} \oplus \Gamma_{t+1}^{a,z_2} \oplus \cdots \oplus \Gamma_{t+1}^{a,z_{|Z|}}$$

其中笛卡尔和⊕定义为：

$$A \oplus B = \{\alpha+\beta \mid \forall\alpha\in A, \forall\beta\in B\}$$

最后得到所有动作向量集合：

$$\Gamma_{t+1} = \bigcup_{a\in A}\Gamma_{t+1}^a$$

反复 update 至Γ_n收敛即可精确求解 POMDP 问题. 每次 update 的计算复杂度近似为 $O(|S|^2|A||\Gamma_t|^{|Z|})$[17]，因而精确求解存在着历史灾和维度灾的问题. 虽然 Witness 算法和增量裁剪算法等对精确算法有所改进，但在极端情况下计算复杂度还是不能降低.

2.3 基于点的 POMDP 近似求解

对于大部分的 POMDP 问题，Agent 所能到达的信念点集合 B 往往只是信念空间的一小部分，因此可以用基于点的算法来求得其误差在一定范围之内的近似解，避免精确求解中计算笛卡尔和的巨大计算量，通过增加迭代次数保证算法效果.

基于点进行 backup 和精确算法的 update 的比较如图 1 所示. 精确求解算法在整个信念空间上进行，所以无法先行确定动作 a 之后各个观察下的最优向量，只能选取所有可能的向量作笛卡尔和，因而计算量很大. 基于点的方法中，执行动作 a 之后的每个观察下的最优向量都可以先行确定，从而可以根据$|Z|$个观察所对应的最优向量计算出执行动作 a 的回报值，再比较得出回报值最高的最优动作，最后通过 backup 操作得到 $\boldsymbol{b}$ 在一次更新后的最优向量.

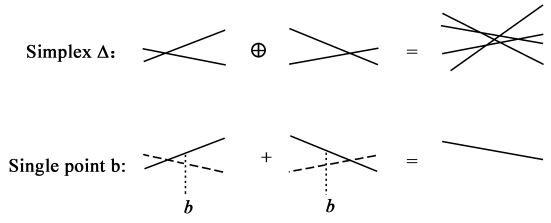

图1 单个信念点上值迭代与精确值迭代的差别

在点集 B 上由Γ_t构建Γ_{t+1}过程如下：

$$\Gamma_{t+1,\boldsymbol{b}}^a = \Gamma_1^a + \sum_{z\in Z}\arg\max_{\alpha\in\Gamma_{t+1}^{a,z}}[\sum_{s\in S}\alpha(s)\boldsymbol{b}(s)]$$

$$\Gamma_{t+1,\boldsymbol{b}} = \cup_{a\in A}\Gamma_{t+1,\boldsymbol{b}}^a$$

$$\text{backup}(\boldsymbol{b}) = \arg\max_{\alpha\in\Gamma_{t+1,\boldsymbol{b}}}\sum_{s\in S} b(s)\alpha(s)$$

$$\Gamma_{t+1} = \bigcup_{\boldsymbol{b}\in B}\text{backup}(\boldsymbol{b})$$

在点集 B 上进行一次 backup 的计算复杂度近似为 $O(|S|^2|A||Z\|B|^2)$. 基于点的方法在达到终止条件之前反复执行两个步骤：探索新的信念点来扩张信念点集合 B；在 B 上更新值函数 Γ. 各种基于点的值迭代方法的主要差别在于不同的信念点集探索方法[18].

2.4 最优策略下的可达区域

基于点的算法的核心思想是可到达区域的概念. 可到达区域 $R(b_0)$是从初始信念点 b_0经过任意动作和观察序列能够到达的信念点集合[8]. 但第 t 步时 $R(b_0)$中增加信念点的数量级为$(|A\|Z|)^t$，随着步数 t 的增加 $R(b_0)$的规模也较为可观. $R^*(b_0)$是从 b_0开始按照最优策略所到达信念点的集合[19]，第 t 步时 $R^*(b_0)$中增加信念点的数量级为$|Z|^t$. 如图 2 所示，$R^*(b_0)$的规模远小于 $R(b_0)$，因而在较大规模的问题中基于 $R^*(b_0)$采样更加高效.

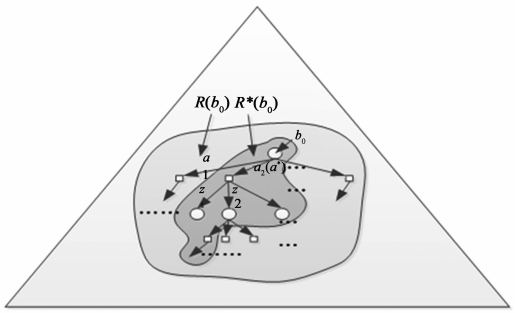

图2 信念空间Δ包含可到达区域$R(b_0)$, $R(b_0)$

尽管 $R^*(b_0)$规模相对较小，但足以用于计算出 b_0处的最优策略[19]. 然而最优策略无法预知，所以一般通过启发式的方法来对 $R^*(b_0)$进行近似.

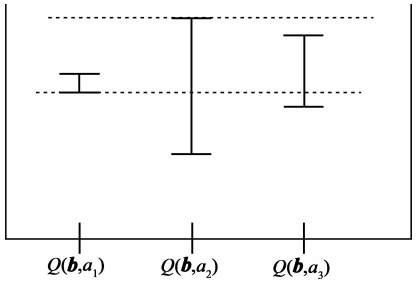

图3 选择最优动作的不同标准

已有的基于点的近似算法在探索 $R^*(b_0)$时尝试了不同的选择最优动作的标准. 如图 3 所示，信念点 $\boldsymbol{b}$ 处有 3 个可供选择的动作 a_1、a_2、a_3，其动作值函数 $Q(\boldsymbol{b}, a_i)$分别在各自的下界和上界之间取值. 在此例中 PEMA 等算法根据动作值函数下界的最大值会选取动作 a_1作为最优策略；HSVI 等算法根据动作值函数上界选

择动作 a_2作为最优策略.

3　PBVIOP 算法

3.1　算法思想

目前已有的 $R^*(b_0)$近似算法仍有改进的空间. PEMA 算法仅根据值函数下界选取最优动作,则值函数下界取值较高的信念点更可能会被探索到,然后在该点上的 backup 操作又只会使得该点附近区域的值函数下界会有所提升而其他信念区域的值函数下界几乎没有提升,从而在下一次的探索中该点附近区域的信念点又会被优先探索到,因此算法不能保证值函数收敛到全局最优解. HSVI 等算法根据 IE-MAX 原则只根据值函数上界值最大来选择动作,上界在更新中不断降低,因而即使在某次迭代中只是找到了次优动作也不会影响值函数最终能够收敛到全局最优. 但值函数的上界通过线性规划或 sawtooth 算法[10]来近似计算,其收敛速度非常缓慢,HSVI 等算法虽然在理论上保证收敛,但在选择最优动作时完全不考虑迭代收敛效率较高的值函数下界,影响了整个算法的收敛效率,不利其应用于大规模的 POMDP 问题.

事实上动作值函数在上界和下界之间取值,单单以上界或下界的值来评估动作值函数都是片面的. 在图3 的示例中,以 $Q(\boldsymbol{b},a_i)$的上界和下界为端点的整个线段反映了 $Q(\boldsymbol{b},a_i)$的取值情况,仅仅以线段的上端点或下端点来评价 $Q(\boldsymbol{b},a_i)$显然不够全面. 事实上就整个线段比较而言,在图3 的示例中可能选择 a_3作为最优动作更为合理,尽管 $Q(\boldsymbol{b},a_3)$的上界和下界都不是最大值,但是 $Q(\boldsymbol{b},a_3)$值最大的概率可能最大.

本文提出了选择最优动作的新标准:以所有动作的函数值在其上界和下界之间的概率分布为基础,计算每个动作的值函数取值最大的概率,再选择概率值最大的动作. 基于新标准选择动作更加合理,可以更准确地探索到 $R^*(b_0)$附近的区域,从而提高迭代效率.

3.2　基于蒙特卡罗的概率计算

动作 a_i的值函数 $Q(\boldsymbol{b},a_i)$的取值可表示为介于上界 $\overline{Q}(\boldsymbol{b},a_i)$与下界$\underline{Q}(\boldsymbol{b},a_i)$之间的随机变量 x_i,设其概率密度函数为$p_i(x_i)$,则 $\int_{\underline{Q}(a_i)}^{\overline{Q}(a)} p_i(x_i)\,\mathrm{d}\,x_i=1$. 所有动作的联合概率密度函数 $p(y)$可以表示为:

$$p(\boldsymbol{y})=p(x_1,x_2,\cdots,x_n)$$

其中 $\boldsymbol{y}$ 是一个 n 维向量:$\boldsymbol{y}=(x_1,x_2,\cdots,x_n)$满足 $\oint_{\Omega} p(x_1,x_2,\cdots,x_n)\,\mathrm{d}\,x_1\mathrm{d}\,x_2\cdots\mathrm{d}\,x_n=1$. 其中

$$\boldsymbol{\Omega}=\left\{(x_1,x_2,\cdots,x_n)\left|\begin{array}{c}\underline{Q}(\boldsymbol{b},a_1)\leqslant x_1\leqslant\overline{Q}(\boldsymbol{b},a_1)\\ \underline{Q}(\boldsymbol{b},a_2)\leqslant x_2\leqslant\overline{Q}(\boldsymbol{b},a_2)\\ \vdots\\ \underline{Q}(\boldsymbol{b},a_n)\leqslant x_n\leqslant\overline{Q}(\boldsymbol{b},a_n)\end{array}\right.\right\}$$

则动作 a_i的值函数的取值 x_i最大的概率为:

$$F^*(a_i)=P(x_i>x_j,\forall j\neq i)$$

$$=\oint_{\Omega_i} p(x_1,\cdots,x_n)\,\mathrm{d}\,x_1\cdots\mathrm{d}\,x_n$$

$$\Omega_i=\Omega\cap\{(x_1,x_2,\cdots,x_n)\mid x_i>x_j,\forall j\neq i\}$$

由于Ω_i是 n 维空间的一个封闭区域,$F^*(a_i)$的计算涉及高维积分. 随着维数 n 的增加,计算难度和复杂度将大大增加,本文通过蒙特卡罗法来求其近似值.

定理 1　设$y_1,y_2,\cdots,y_m$为 Ω 上按概率密度 $p(\boldsymbol{y})$选取的 m 个随机样点,其中满足$y_k\in\Omega_i$的随机样点个数为 n_i,则$F^*(a_i)\approx\dfrac{n_i}{m}$.

证明:构造两个函数$Q_i(\boldsymbol{y})$和$F_i(\boldsymbol{y})$:

$$Q_i(\boldsymbol{y})=\begin{cases}p(y), & y\in\Omega_i\\ 0, & \text{else}\end{cases},\quad F_i(\boldsymbol{y})=\begin{cases}\dfrac{Q_i(\boldsymbol{y})}{p(y)}, & p(\boldsymbol{y})\neq 0\\ 0, & p(\boldsymbol{y})=0\end{cases}$$

则:$F^*(a_i)=\oint_{\Omega} Q_i(\boldsymbol{y})\mathrm{d}y=\oint_{\Omega} F_i(\boldsymbol{y})p(\boldsymbol{y})\mathrm{d}\boldsymbol{y}$

由此 $F^*(a_i)$即随机变量 $F_i(\boldsymbol{y})$的数学期望值,由于$y_1,y_2,\cdots,y_m$为 Ω 上按概率密度 $p(\boldsymbol{y})$选取的随机样点,可求 $F_i(\boldsymbol{y})$的数学期望近似值.

$$F^*(a_i)\approx\frac{1}{m}\sum_{j=1}^{m}F_i(y_j)=\frac{1}{m}\sum_{j=1}^{m}\frac{Q_i(y_j)}{p(y_j)}$$

当$y_j\in\Omega_i$时,$\dfrac{Q_i(y_j)}{p(y_j)}=\dfrac{p(y_j)}{p(y_j)}=1$,其它情况下$\dfrac{Q_i(y_j)}{p(y_j)}=0$. 由于 m 个随机样点中满足$y_k\in\Omega_i$的随机样点个数为 n_i,因此$F^*(a_i)\approx\dfrac{1}{m}\sum_{j=1}^{m}\dfrac{Q_i(y_j)}{p(y_j)}=\dfrac{n_i}{m}$ 证毕.

本文参照 AEMS1 算法[14]假定动作的最优值函数在上下界之间均匀分布,对动作值函数进行取样,并由此计算动作最优的概率.

3.3　PBVIOP 算法

PBVIOP 算法(算法 1)初始化值函数的上下界之后,反复调用子函数 PBVIOPExplore 从 b_0出发进行深度探索并更新值函数的上界和下界,直至 b_0处取值收敛为止.

算法 1　PBVIOP

Input:POMDP

Output:$\underline{V},\pi_{\underline{V}}(\boldsymbol{b}_0)$

$\underline{V}\leftarrow$Blind Policy( )

$\overline{V}\leftarrow\leftarrow$FIB( )

While $(\overline{V}(b_0)-\underline{V}(b_0))>\varepsilon$ do

PBVIOPExplore(b_0,0)

end while

本文以向量集$\underline{V}$表示下界,信念点 $\boldsymbol{b}$ 处的下界取值

$\underline{V}(\boldsymbol{b}) = \max_{\alpha \in \underline{V}} \boldsymbol{b} \cdot \boldsymbol{\alpha}$. 下界$\underline{V}$由盲目策略来初始化,通过backup来更新. 本文用多个信念点-值对$(b_i, \bar{v}_i)$所组成的集合$\overline{V}$表示上界,$\overline{V}$由FIB算法初始化,以加入新的信念点-值对来更新,表示为:

$$\overline{V}' = \overline{V} \cup (\boldsymbol{b}, \max_{a \in A} \overline{Q}(\boldsymbol{b}, a)),$$

$$\overline{Q}(\boldsymbol{b}, a) = \sum_{s \in S} R(s, a) \cdot b(s) + \gamma \sum_{z \in Z} P(z | \boldsymbol{b}, a) \overline{V}(b^{a,z})$$

信念点$\boldsymbol{b}$处的上界取值$\overline{V}(\boldsymbol{b})$由$\boldsymbol{b}$向集合$\overline{V}$形成的凸包投射获得,该投射可由线性规划来精确计算,本文则通过sawtooth算法求其近似值. 子函数PBVIOPExplore(算法2)从当前信念点$\boldsymbol{b}$出发,根据概率最大准则选择最优的动作a^*,再以上界和下界之差概率加权最大为依据选择观察z^*,从而探索到新的信念点b^{a^*,z^*};再进行迭代,直至某次探索到的信念点上的当前值函数上下界的差值小于阈值ε/γ^t为止. 然后对探索到的信念点集B按照信念点被探索到的顺序逆序更新值函数上界和下界,完成一次深度探索.

算法2 PBVIOPExplore 子函数

Input: $\boldsymbol{b}, t$

Output: $\underline{V}, \overline{V}$

if $(\overline{V}(\boldsymbol{b}) - \underline{V}(\boldsymbol{b})) < \frac{\varepsilon}{\gamma^t}$ then

  return

else

  for $k = 1$ to i MaxRound do

    for $i = 1$ to $|A|$ do

      $Q(\boldsymbol{b}, a_i) = (\overline{Q}(\boldsymbol{b}, a_i) - \underline{Q}(\boldsymbol{b}, a_i)) * x + \underline{Q}(\boldsymbol{b}, a_i)$

    //基于$Q(\boldsymbol{b}, a_i)$ $Q(\boldsymbol{b}, a_i)$的分布函数随机取值

    end for

    $i = \text{argmax}_i \text{argmax}_i Q(\boldsymbol{b}, a_i)$ $Q(\boldsymbol{b}, a_i)$

    i $\text{count}_i = \text{count}_i + 1$

  end for

  $iMaxindex = \text{argmax}_i(count_i)$

  $a^* \leftarrow a_{i\text{Maxindex}}$

  $z^* \leftarrow \text{argmax}_z(P(z | \boldsymbol{b}, a^*)(\overline{V}(b^{a^*,z}) - \underline{V}(b^{a^*,z})))$

  PBVIOPExplore$(b^{a^*,z^*}, t+1)$

  $\underline{V} = \text{backup}(\boldsymbol{b}, \underline{V})$

  $\overline{V} = \text{sawtooth}(\boldsymbol{b}, \overline{V})$

end if

PBVIOP算法中一次选择最优动作时会比HSVI算法多出求$\underline{Q}(\boldsymbol{b}, a_i)$和$P^*(a_i)$近似值的计算,其探索一个信念点的计算复杂度约为:$O(|A| i\text{Maxround}) + |Z| |S|(|S| + |\overline{V}| + |\underline{V}| + |A||\underline{V}|)$.

PBVIOP算法在选择最优动作时同时考虑了最优动作值函数的上界和下界. 在迭代过程中下界持续上升而上界会持续下降,随着值函数上下界之差逐渐缩小,对各个动作最优概率的估算会更加精确,因而保证了值函数的收敛. 因为算法同时更新值函数的上界和下界,并以值函数在上界和下界之间的分布来计算动作最优的概率,所以在信念点上更新值函数的上界和下界不会增加该点以后被探索到的可能性,故而算法会收敛到全局最优解.

4 实验

4.1 实验设置

本文实验对比了PBVIOP算法、HSVI算法和GapMin算法运算情况,因为PBVIOP算法和HSVI算法的主要差别在于最优动作的选择,而GapMin算法是目前最高效的POMDP规划算法之一. 本文在常见4个数据集上进行实验,其中Tiger、Hallway是早期的经典迷宫问题;RockSample模拟了Agent采样矿石的科学考察任务,是一个可扩展的问题[10]. 实验所用数据集的状态、动作和观察规模如下表:

表1 POMDP标准数据集的规模

问题	状态数$\|S\|$	动作数$\|A\|$	观察数$\|Z\|$
Hallway	61	5	21
Tiger-grid	36	5	17
RockSample(5,5)	801	10	2
RockSample(7,8)	12545	13	2

本文实验中复用了Guy Shani教授提供的POMDP-Solver部分代码. 对每个问题设定折扣因子为0.95,分别用PBVIOP算法、HSVI算法和GapMin算法各做10次运算,再对10次运算的结果取平均值,选取运算时间和平均折扣回报值(Average Discounted Reward, ADR)作为评价指标. 平均折扣回报值表示了生成策略的质量,由生成的策略模拟运行100步计算得出折扣回报值,通过反复500次的模拟来计算平均折扣回报值.

4.2 实验结果分析

实验结果如表2所示,可见大多数情况下PBVIOP算法有较好的收敛效果.

图4是HSVI、GapMin和PBVIOP在四个问题上实验结果的详细对比,表示了生成策略的平均折扣回报值的演变情况. 图中横坐标为算法运行时间(s),纵坐标为ADR值;实线表示HSVI算法对应的结果,短划线表示GapMin算法对应的结果,圆点线表示PBVIOP算法对应的结果.

表2 实验结果数据

问题	ADR			Time(s)		
	HSVI	GapMin	PBVIOP	HSVI	GapMin	PBVIOP
Hallway	0.52	0.526	0.52	325	465	103
Tiger-grid	1.64	1.656	1.64	150	546	110
RockSample (5,5)	19.65	20.033	19.66	88	2356	15
RockSample (7,8)	16.35	21.02	21.22	2448	2635	1584

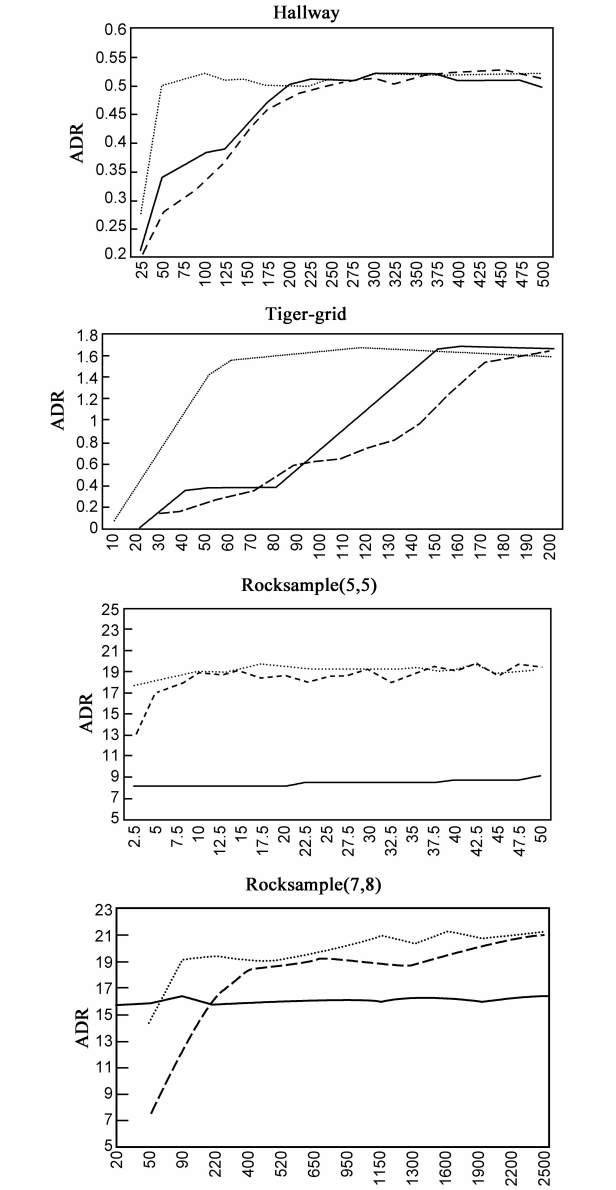

图4　HSVI、GapMin和PBVIOP在4个不同标准数据集上运行结果的对比

在求解 Hallway 和 Tiger-grid 问题的实验中,因为问题规模较小,PBVIOP 算法和 HSVI 算法收敛到相同的 ADR,GapMin 算法的 ADR 略高一点. 而 PBVIOP 算法的收敛效率明显较高,在 Hallway 问题求解中比 HSVI 算法快 3.15 倍,比 GapMin 算法快 4.51 倍;在 Tiger-grid 问题求解中比 HSVI 算法快 1.36 倍,比 GapMin 算法快 4.96 倍.

在求解 RockSample(5,5)问题的实验中,PBVIOP 算法收敛到的 ADR 比 HSVI 算法高出较多,收敛效率比 HSVI 算法快 5.86 倍. PBVIOP 算法收敛到的 ADR 略低于 GapMin 算法,但其收敛效率比 GapMin 算法快 157.06 倍.

在求解 RockSample(7,8)问题的实验中,PBVIOP 算法和 GapMin 算法收敛到的 ADR 都比 HSVI 算法高出较多,且 PBVIOP 算法收敛到的 ADR 比 GapMin 算法略高. PBVIOP 算法收敛效率比 HSVI 算法快 1.54 倍,比 GapMin 算法快 1.66 倍.

虽然 GapMin 算法和 HSVI 算法一样选择值函数上界最优的动作,但 GapMin 算法在每轮迭代中会探索所有 Gap 大于当前阈值的信念点,因而 GapMin 算法可以更加有效地降低上界值,在状态规模不太大的 POMDP 问题上找到全局最优解. 但随着 POMDP 问题中状态数的增加,上界的下降效果变差,GapMin 算法也难以有效地求解 POMDP 问题. 另外由于 GapMin 算法多探索了许多信念点,其收敛效率受到较大影响.

实验结果表明 PBVIOP 算法比 HSVI 和 GapMin 算法有更高的收敛效率,并且随着 POMDP 问题规模的增加,其收敛到的 ADR 也会明显地优于 HSVI 算法,和 GapMin 算法相当. 随着状态数目的增加,上界的下降速度会显著降低,因而 HSVI 和 GapMin 算法的收敛效率直接受到了影响. 另一方面,随着动作数量的增加,PBVIOP 算法探索的 $R^*(b_0)$ 和 HSVI 算法探索的 $R^*(b_0)$ 会有更大的差异,因而 PBVIOP 算法的效果会更优于 HSVI 算法. 这说明与单纯利用上界相比而言,同时利用上下界能够更快更优地探索到 $R^*(b_0)$ 附近的区域,对于算法性能和收敛质量的提升有很大的帮助.

5　结束语

本文提出了一种基于概率的最优策略值迭代方法 PBVIOP,解决了启发式探索最优策略可达区域 $R^*(b_0)$ 时需要保障值函数上下界收敛效率的问题. PBVIOP 算法与现有基于点的值迭代算法不同之处在于使用一种有效的新方法来探索最优策略可达区域 $R^*(b_0)$. PBVIOP 算法同时维持值函数的上界和下界,在启发式的深度探索中,用蒙特卡罗法估算各个动作值函数最优的概率,选择概率最大的动作为最优策略,再贪婪探索不确定性最大的后继信念点. 实验结果表明,与 HSVI 和 GapMin 算法相比,PBVIOP 算法在基准数据集上有更高的收敛效率并能够获得较优的策略. 未来的工作一方面是在 APPL 平台上实现本算法,完善实验配置,尝试和 HHOP 等算法进行比较分析以完善本算法;另一方面是进一步优化值函数的概率分布模型和后继信念点的选择标准,并尝试每步探索多个有效的信念点来近似最优策略可达区域,从而进一步提高一次深度探索的效率.

参考文献

[1] S Russell, PNorvig. Artificial Intelligence: A Modern Approach[M]. Prentice-Hall, 1995.

[2] T Smith. Probabilistic planning for robotic exploration[D]. Massachusetts Institute of Technology, 2007.

[3] J D Williams, S Young. Partially observable Markov decision processes for spoken dialog systems [J]. Computer Speech & Language, Elsevier, 2007, 21(2): 393 - 422.

[4] 赵二虎,阳小龙,等. CPSM:一种增强 IP 网络生存性的客户端主动服务漂移模型[J]. 电子学报, 2010, 38(9): 2134 - 2139.
Zhao Er-hu, Yang Xiao-long, et al. CPSM: Client-side proactive service migration model for enhancing IP network survivability [J]. Acta Electronica Sinica, 2010, 38(9): 2134 - 2139. (in Chinese)

[5] 张子宁,单甘霖,段修生. 基于部分可观马氏决策过程的多平台主被动传感器调度[J]. 电子学报, 2014, 42(10): 2104 - 2109.
Zhang Zi-ning, Shan Gan-lin, Duan Xiu-sheng. POMDP-based scheduling of active/passive sensors in multi-platform [J]. Acta Electronica Sinica, 2014, 42(10): 2104 - 2109. (in Chinese)

[6] M Hauskrecht. Value-function approximations for partially observable Markov decision processes [J]. Journal of Artificial Intelligence Research, 2000, 13(1): 33 - 94.

[7] 刘海涛,洪炳熔,等. 不确定性环境下基于进化算法的强化学习[J]. 电子学报, 2006, 34(7): 1356 - 1360.
Liu Hai-tao, Hong Bing-rong, et al. Evolutionary algorithm based reinforcement learning in the uncertain environments [J]. Acta Electronica Sinica, 2006, 34(7): 1356 - 1360. (in Chinese)

[8] Pineau J, Gordon G, Thrun S. Point-based value iteration: An anytime algorithm for POMDPs [A]. International Joint Conference on Artificial Intelligence [C]. Acapulco, Mexico: Morgan Kaufmann, 2003. 1025 - 1032.

[9] J Pineau, G Gordon. POMDP planning for robust robot control [A]. International Symposium on Robotics Research [C]. San Francisco, USA: Springer, 2005, 69 - 82.

[10] T Smith, R G Simmons. Point-based POMDP algorithms: Improved analysis and implementation [A]. Conference on Uncertainty in Artificial Intelligence [C]. Edinburgh, United kingdom: AUAI Press, 2005, 542 - 547.

[11] H Kurniawati, D Hsu, W S Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces [A]. Robotics: Science and Systems [C]. Zurich, Switzerland: MIT Press, 2008, 65 - 72.

[12] P Poupart, K E Kim, D Kim. Closing the gap: Improved bounds on optimal POMDP solutions [A]. International Conference on Planning and Scheduling [C]. Freiburg, Germany: AAAI Press, 2011. 194 - 201.

[13] Z Zhang, D Hsu, W S Lee. Covering Number for Efficient Heuristic-based POMDP Planning [A]. International Conference on Machine Learning [C]. Beijing, China: International Machine Learning Society, 2014. 48 - 60.

[14] S Ross, B Chaib-Draa. AEMS: An anytime online search algorithm for approximate policy refinement in large POMDPs [A]. International Joint Conference on Artificial Intelligence [C]. Hyderabad, India: Morgan Kaufmann, 2007. 2592 - 2598.

[15] 章宗长,陈小平. 杂合启发式在线 POMDP 规划[J]. 软件学报, 2013, 24(7): 1589 - 1600.
Zhang Zong-zhang, Chen Xiao-ping. Hybrid heuristic online planning for POMDPs [J]. Journal of Software, 2013, 24(7): 1589 - 1600. (in Chinese)

[16] L P Kaelbling. Learning in Embedded Systems [M]. MIT Press, 1993.

[17] R D Smallwood, E J Sondik. The optimal control of partially observable markov processes over a finite horizon [J]. Operations Research, 1973, 21(5): 1071 - 1088.

[18] G Shani, J Pineau, R Kaplow. A survey of point-based POMDP solvers [J]. Autonomous Agents and Multi-Agent Systems, 2013, 27(1): 1 - 51.

[19] D Hsu, W S Lee, N Rong. What makes some POMDP problems easy to approximate? [A]. Advances in Neural Information Processing Systems [C]. Vancouver, BC, Canada: Curran Associates Inc, 2007. 689 - 696.

作者简介

刘 峰 男,1976 年生于江苏泰州. 南京大学软件学院讲师. 研究方向为强化学习、智能规划.
E-mail: ufeng_nju@163.com

王崇骏 男,1975 年生于江苏盱眙,南京大学计算机科学与技术系教授,中国计算机学会高级会员. 研究方向为 Agent 及多 Agent 系统、复杂网络分析及智能应用系统.

骆 斌 男,1967 年生,南京大学软件学院教授,博士生导师,中国计算机学会杰出会员. 研究方向为软件工程、人工智能.