软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(8):3086−3102 [doi: 10.13328/j.cnki.jos.006318]

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

基于受限 MDP 的无模型安全强化学习方法*

朱 斐[1,2,3], 葛洋洋[1], 凌兴宏[1], 刘 全[1]

[1](苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

[2](软件新技术与产业化协同创新中心, 江苏 南京 210093)

[3](江苏省计算机信息处理技术重点实验室(苏州大学), 江苏 苏州 215006)

通信作者: 朱斐, E-mail: zhufei@suda.edu.cn

摘 要: 很多强化学习方法较少地考虑决策的安全性, 但研究领域和工业应用领域都要求的智能体所做决策是安全的. 解决智能体决策安全问题的传统方法主要有改变目标函数、改变智能体的探索过程等, 然而这些方法忽略了智能体遭受的损害和成本, 因此不能有效地保障决策的安全性. 在受限马尔可夫决策过程的基础上, 通过对动作空间添加安全约束, 设计了安全 Sarsa(λ)方法和安全 Sarsa 方法. 在求解过程中, 不仅要求智能体得到最大的状态-动作值, 还要求其满足安全约束的限制, 从而获得安全的最优策略. 由于传统的强化学习求解方法不再适用于求解带约束的安全 Sarsa(λ)模型和安全 Sarsa 模型, 为在满足约束条件下得到全局最优状态-动作值函数, 提出了安全强化学习的求解模型. 求解模型基于线性化多维约束, 采用拉格朗日乘数法, 在保证状态-动作值函数和约束函数具有可微性的前提下, 将安全强化学习模型转化为凸模型, 避免了在求解过程中陷入局部最优解的问题, 提高了算法的求解效率和精确度. 同时, 给出了算法的可行性证明. 最后, 实验验证了算法的有效性.

关键词: 受限马尔可夫决策过程; 安全强化学习; 多维约束; Sarsa(λ)算法; Sarsa 算法

中图法分类号: TP18

中文引用格式: 朱斐, 葛洋洋, 凌兴宏, 刘全. 基于受限 MDP 的无模型安全强化学习方法. 软件学报, 2022, 33(8): 3086–3102. http://www.jos.org.cn/1000-9825/6318.htm

英文引用格式: Zhu F, Ge YY, Ling XH, Liu Q. Model-free Safe Reinforcement Learning Method Based on Constrained Markov Decision Processes. Ruan Jian Xue Bao/Journal of Software, 2022, 33(8): 3086−3102 (in Chinese). http://www.jos.org.cn/1000-9825/6318.htm

Model-free Safe Reinforcement Learning Method Based on Constrained Markov Decision Processes

ZHU Fei[1,2,3], GE Yang-Yang[1], LING Xing-Hong[1], LIU Quan[1]

[1](School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

[2](Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210093, China)

[3](Provincial Key Laboratory for Computer Information Processing Technology (Soochow University), Suzhou 215006, China)

Abstract: Many reinforcement learning methods do not take into consideration the safety of decisions made by agents. In fact, regardless of many successful applications in research and industrial area, it is still necessary to make sure that agent decisions are safe. The traditional approaches to address the safety problems mainly include changing the objective function, changing the exploration process of agents and so on, which, however, neglect the possible grave consequences caused by unsafety decisions and, as a result, cannot effectively solve the problem. To address the issue, a safe Sarsa(λ) and a safe Sarsa method, based on the constrained Markov decision processes, are proposed by imposing safety constraints to the action space. During the solution process, the agent should not only seek to get the maximum state-action value, but also satisfy the safety constraints, so as to obtain an optimal safety strategy. Since the standard reinforcement learning methods are no longer suitable for solving the safe Sarsa(λ) and safe Sarsa model, in order to obtain the global

* 基金项目: 国家自然科学基金(61303108, 61772355); 江苏省高校自然科学研究项目(17KJA520004); 苏州市重点产业技术创新-前瞻性应用研究项目(SYG201804); 高校省级重点实验室(苏州大学)项目(KJS1524); 江苏高校优势学科建设工程资助项目
收稿时间: 2019-08-30; 修改时间: 2020-09-08; 采用时间: 2021-02-03

optimal state-action value function under the constrained conditions, a solution model of safe reinforcement learning is also introduced. Such model is based on linearized multidimensional constraints and adopts the Lagrange multiplier method to transform safe reinforcement learning model into a convex model provided that the objective and constraint functions are differentiable. The proposed solution algorithm guides the agent away from a local optimal and improves the solution efficiency and precision. The feasibility of the algorithm is proved. Finally, the effectiveness of the algorithm is verified by experiments.

Key words: constrained Markov decision processes; safe reinforcement learning; multiple constraints; Sarsa(λ) algorithm; Sarsa algorithm

强化学习(reinforcement learning, RL)通过与动态环境不断交互, 利用“试错式”的学习, 寻求回报最大的策略[1]. 强化学习方法得到了研究人员和产业界的广泛关注, 在多个领域积累了较多的研究成果和应用[2−5]. 但是, 目前大多数强化学习方法并没有充分地控制风险、保障决策的安全性, 甚至很多方法特意加入了带有随机性质的探索性学习, 以期望获得更好的近似最优解. 如果在现实世界的任务中直接应用强化学习的方法, 让智能体(agent)进行不受约束地探索, 很可能会使系统陷入危险状态, 使智能体及控制的物体受到损害. 更有研究人员发现, 传统方法所学的最优化准则未必适用于在执行过程中会遇到危险状态的任务[6].

安全强化学习(safe reinforcement learning, SRL)是指在满足安全约束的情况下, 通过最大化期望回报值得到最优策略[7]. 现有安全强化学习方法主要包括改变优化准则和改进智能体的探索过程[8]. 基于改变优化准则的安全强化学习方法是通过改变优化的目标函数来保证智能体的安全, 使智能体在面对最坏情况时能够合理选择产生最大回报值的最优策略[9]. 这种标准可以降低风险或不良情况发生的可能性[10]. 改变优化准则的方法通过设置相关风险的参数来提高智能体的风险敏感性. 在该方法中, 最优化准则采用指数效用函数[11]或回报值和风险的线性结合的形式[12], 风险被定义为回报值的方差[13]或智能体进入危险状态的概率[14]. 对智能体探索过程进行改造的安全强化学习方法通过对状态空间和行为空间实施随机探索来获得对任务的认知[15], 因此只有当信息足够多时, 算法才会得到改善. 主要有两种方法来修改智能体的探索过程以避免危险情况, 第 1 种方法通过对外部信息的整合来改进算法[16], 第 2 种方法使用风险导向的探索来提高安全性[17].

强化学习领域已经积累了很多研究成果. 在从研究转向实际应用的过程中, 还需要保证决策的安全性. 在安全强化学习的研究中, Chow 等人建立了联合风险控制和决策控制的目标[18]; Bušić 等人引入了 Kullback-Leibler 代价限制动作[19]. 这些方法通过改变目标函数来限制智能体的动作, 以达到避开危险动作的目的, 但都忽略了危险状态给智能体带来伤害时所造成的损失. Alshiekh 等人加入了“屏蔽层”来监视动作并修正破坏规则的动作[20]; Fulton 等人提出了形式证明加上运行时监控, 以保证智能体的安全性[21]; Chow 等人将控制理论中经典的李雅普诺夫函数(Lyapunov function)引入强化学习, 缓解了智能体无限制的探索所带来的问题[22]; Cheng 等人在无模型强化学习的基础上增加了基于模型的控制屏障函数和未知系统动力学的在线学习, 保障学习时的安全[23]. 这些方法通过添加约束来保证智能体的安全, 但是没有系统地描述和解决问题, 且表现令人难以满意. 将智能体的危险问题作为一种约束限制, 是保证智能体安全的一种有效方法.

Achiam 等人提出的受限策略最优化(constrained policy optimization, CPO)方法可以用来解决模拟机器人运动任务中的安全约束问题, 但该方法只能近似满足约束条件, 没有提出可行的方法求解满足约束条件的最优解[24]; Tessler 等人提出了回报值受限的策略优化方法(reward constrained policy optimization, RCPO), 用来解决强化学习中可能存在奖励信号漏洞和错误设定等问题[25]; Miryoosefi 等人提出了带凸约束的基于可接近的策略优化强化学习方法, 用来解决强化学习中存在的不安全动作, 或在奖励值稀疏的情况下, 近似专家轨迹等约束问题[26]; Hasanbeig 等人提出了带有逻辑约束的谨慎强化学习, 用来解决强化学习中存在危险状态的问题[27]. 上述受限的强化学习方法可以近似解决强化学习中的安全等限制问题, 为保证智能体的安全提供了新的求解思路. 然而, 这些方法仍然存在诸多不足, 如求解方案不完善、未能系统地描述和解决智能体的安全问题等.

人工智能必须保证所控制物体的安全, 降低导致危险的风险. 要完成这一点, 就需要改变传统的建模方法, 建立有效融合安全约束的适合模型并解决一系列相关问题. 针对强化学习中智能体的安全问题, 本文提

出了基于受限马尔可夫决策过程(constrained Markov decision processe, CMDP)[28]的无模型安全强化学习方法, 分别在经典的 Sarsa(λ)算法和 Sarsa 算法的基础上实现了安全 Sarsa(λ)方法(safe Sarsa(λ) method)和安全 Sarsa 方法(safe Sarsa method). 安全 Sarsa(λ)方法使用受限马尔可夫决策过程来描述, 它在引入资格迹概念的 Sarsa(λ)的 Q 值更新基础上增加了多维安全约束函数, 通过约束, 将状态空间和动作空间分别分为可行空间和不可行空间, 并采用拉格朗日乘数法进行求解, 保证智能体的安全. 为了进一步分析资格迹λ对本文提出的安全算法的影响, 本文在强化学习方法 Sarsa 方法的基础上增加了多维安全约束函数, 设计了安全 Sarsa 方法. 在智能体的运行过程中, 满足约束条件的安全状态和安全动作才是智能体的可行状态和可行动作, 从而避免智能体执行危险动作而进入危险状态, 造成不必要的损害. 本文提出的安全强化学习方法可以应用在机器人技术、自动驾驶等对安全要求较高的任务中.

1 相关工作

1.1 强化学习

在强化学习任务中, 智能体通过与环境交互感知环境的状态并选择执行相应的动作获得回报值, 智能体行动的目的是最大化长期累积回报值[29]. 马尔可夫决策过程是解决大部分强化学习问题的框架, 用四元组(S,A,P,R)描述, 其中, S 是智能体的状态空间集, 其元素个数有限; A 是动作空间集; P 是状态转移概率; R 是回报函数[30]. 马尔可夫决策过程中的状态转移概率包含了动作, 状态转移概率公式为

$$P_{ss'}^{a}=P[s_{t+1}=s'\mid s_t=s,a_t=a] \tag{1}$$

在 t 时刻, 智能体处于当前状态 s_t, 根据策略π选择动作 a_t, 环境根据智能体执行的动作给智能体一个反馈, 使智能体进入下一状态 s_{t+1}, 并获得回报值 r_t. 策略π分为确定性策略和非确定性策略, 表示从状态 s_t 到动作 a_t 的映射. 强化学习中长期累积折扣奖赏定义为

$$R_t=\sum_{t'=t}^{T}\gamma^{t'-t}r_{t'} \tag{2}$$

其中, $\gamma\in(0,1]$是智能体长期收益的折扣因子, R_t 是从时刻 t 到时刻 T 的累积折扣回报值. 强化学习模型示意图如图 1 所示[31].

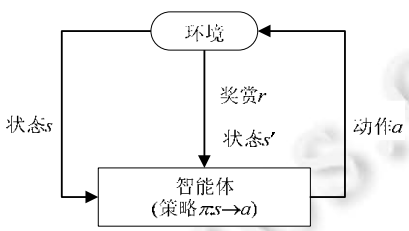

图 1 强化学习模型示意图

状态-动作值函数 $Q^{\pi}(s,a)$在强化学习中用来评估策略的好坏, 是指智能体在当前状态 s_t 下、在策略π的指导下, 执行动作 a_t 获得的累积奖赏之和. 状态-动作值函数为

$$Q^{\pi}(s,a)=E[R_t|s_t=s,a_t=a,\pi] \tag{3}$$

随着迭代步数的不断增加, 状态-动作值收敛到最优, 虽然最优策略不唯一, 但是最优状态-动作值唯一. 最优状态-动作值的计算方法为

$$Q^{*}(s,a)=\max_{\pi}E[R_t|s_t=s,a_t=a,\pi] \tag{4}$$

智能体在当前状态 s_t 下, 在策略π的指导下获取的状态值函数 $V(s)$指的是: 在状态 s_t 处, 采取策略π时, 所有状态-动作值函数的期望. 值函数的计算方法为

$$V^{\pi}(s)=E[R_t|s_t=s,\pi] \tag{5}$$

随着策略的不断迭代, 状态值收敛到最优并且最优值唯一. 最优状态值为

$$V^{*}(s)=\max_{\pi}E[R_t|s_t=s,\pi] \tag{6}$$

1.2 Sarsa(λ)方法和Sarsa方法

Sarsa 算法是一种同策略(on-policy)的时序差分(temporal difference, TD)方法, 是一种无模型的强化学习方法. 强化学习分为有模型的动态规划方法和无模型的强化学习方法, 无模型的强化学习方法包括蒙特卡罗法和时序差分法等[32], Sarsa 算法和 Sarsa(λ)算法都是时序差分法. 同策略是指算法中生成行为策略和更新值函数使用同一策略, 异策略(off-policy)是指算法中生成行为策略和更新值函数使用不同的策略[33]. Sarsa 算法评估值的更新方程为

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha[R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \tag{7}$$

Sarsa 算法中, $Q(s,a)$的值常使用表格进行存储, 不适合解决大规模问题. 在更新状态-动作对值时, 智能体并不实际执行在状态 s_{t+1}的动作 a_{t+1}, 而是将动作 a_{t+1}留到下次迭代时执行. Sarsa 算法是一种单步更新算法, 为更有效地更新所有相关步, Sarsa(λ)算法改单步更新为多步更新, λ指更新获取到回报值之前的所有步数的权重参数, 取值范围为[0,1]. Sarsa(λ)分前向和后向, 前向观点更新 Q 值时需要遍历全部状态序列, 而后向观点是通过引入资格迹 $E(s,a)$更新 Q 值, 其更新公式如下:

$$\delta_t\leftarrow R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t) \tag{8}$$

$$Q(s,a)\leftarrow Q(s,a)+\alpha\delta_t E_t(s,a) \tag{9}$$

上述更新公式(8)和公式(9)中, t 表示时刻, α是学习率, δ_t 表示时间差分误差, a_{t+1} 表示下一状态 s_{t+1} 执行的动作. 当时间步 t 趋于无穷时, 得到最优控制策略[34]. Sarsa(λ)算法使用 TD 误差进行迭代更新, 最终收敛到最优状态-动作值. 公式(9)中, 资格迹 $E_t(s,a)$表示智能体每经历一个状态-动作对(s,a)都会做一个标记, 使 $E(s,a)$的值增加 1.

1.3 受限马尔可夫决策过程

受限马尔可夫决策过程使用五元组(S,A,P,R,C)描述问题, 其中,

- S 是一个状态空间集合, 包含有限的状态数, 元素状态的通用符号是 s. 对于多维度的问题, 可以将一个向量看成一个状态;
- A 是一系列的动作组成的动作空间集合, 用 $A(s)$表示在状态 s 下可用的动作集, a 表示动作;
- P 是转移概率, P'_{sas} 是从状态 s 采取动作 a 到达状态 s'的概率;
- R 是关于即时奖励的一个奖励函数;
- C 表示约束函数集[28].

在受限马尔可夫决策过程的模型中, 目标变化为保证智能体在满足约束的情况下最大化整个情节的奖励函数. 约束函数集合具体表示为

$$C=\{c_i:S\times A\rightarrow\mathbb{R}|i=1,\ldots,k\} \tag{10}$$

其中, k 是一个有限常数, $\mathbb{R}$表示实数集上的约束函数值. 约束函数有等式约束和不等式约束两种表示形式.

传统的强化学习是在马尔可夫决策过程的基础上寻找最优策略, 本文提出的安全 Sarsa 方法和安全 Sarsa(λ)方法是在受限马尔可夫决策过程的基础上寻找最优策略. 基于受限马尔可夫决策过程的强化学习简称受限强化学习(constrained reinforcement learning, CRL).

1.4 拉格朗日乘数法

传统的强化学习方法是通过最大化目标函数使智能体执行最优动作, 忽略了智能体的安全等限制问题. 虽然在目标函数中添加约束可以保证智能体在运行过程中的安全, 但是传统强化学习求解方法难以解决基于约束模型的安全强化学习问题.

本文在传统强化学习求解方法的基础上, 结合拉格朗日乘数法, 求解智能体在当前状态下可以执行的安全最优动作. 其中, 拉格朗日乘数法是求解带有等式约束的最优化方法, 带有不等式约束的最优化问题转化为等式约束最优化问题后, 采用拉格朗日乘数法进行求解, 并采用库恩塔克(Karush Kuhn-Tucker conditions, KKT)条件进行验证. KKT 条件在模型是凸的情况下是充分必要条件, 否则只是必要条件, 用来判定由拉格朗

日乘数法求解得到的解是否是最优解[35]. 带有不等式约束和等式约束的最优化问题可以表示为

$$\left.\begin{aligned}&\max\ f(x_t,u_t)\\&\text{s.t. }\ c_i(x_t,u_t)=C_i,\ i=1,2,...,k'\\&\quad\ c_i(x_t,u_t)\leqslant C_i,\ i=k'+1,...,k\end{aligned}\right\}\tag{11}$$

其中, $c_i(x_t,u_t)=C_i$ 和 $c_i(x_t,u_t)\leqslant C_i$ 是约束函数的抽象表达式, k 是表示所有约束函数数量的常数, k'是表示等式约束函数数量的常数, 非等式约束函数的数量用 $k-k'$表示. 上述最优化问题模型中, 最优解满足$\lambda=0$ 或 $c_i(x_t,u_t)-C_i=0$, $i=k'+1,\ldots,k$. 故当状态 x_t 和动作 a_t 满足严格不等式约束时, 该不等式约束函数不会影响最优化问题的求解, 即约束函数为严格不等式时为无效约束, 只有约束函数为等式约束时是有效约束. 所以将带有不等式约束的最优化问题转化为只带有等式约束的最优化问题, 再用拉格朗日乘数法进行求解可以解决带有不等式约束的最优化问题, 从而降低了最优化问题求解的复杂度.

为简化求解过程, 如果问题是离散的并且状态空间和动作空间不是很大, 可以直接通过约束判定选择安全动作, 保证智能体进入的下一状态是安全的; 如果状态空间和动作空间很大, 甚至是连续的, 可以将模型转换为凸模型, 然后用拉格朗日乘数法进行求解, 模型转化为如下形式:

$$\max L(x_t,u_t)=f(x_t,u_t)-\lambda_{i_1}(c_{i_1}(x_t,u_t)-C_{i_1})-\lambda_{i_2}(c_{i_2}(x_t,u_t)-C_{i_2}),\ i_1\in\{1,2,...,k'\},\ i_2\in\{k'+1,...,k\}\tag{12}$$

在上述模型中, 若模型是非凸的, 满足 $\nabla_{x_t}L(x_t,u_t)=0$ 和 $\nabla_{u_t}L(x_t,u_t)=0$ 的当前状态 x_t 和动作 u_t 是局部最优解. 该局部最优解用梯度下降法求得, 当模型为凸模型时, 最优解满足 KKT 条件, 此时求得的最优解是全局最优解. λ_i 是拉格朗日不定乘子, 代表约束函数变化时, 目标函数的变动. 由于最优解满足约束 $c_i(x_t,u_t)-C_i=0$, 所以λ_i的取值不会影响最优化问题的最终求解[36].

2 基于受限模型的安全 Sarsa(λ)方法和安全 Sarsa 方法

基于受限马尔可夫决策过程的安全强化学习模型使用五元组(S,A,P,R,C)对问题进行描述, 其中, S 是状态空间, 状态元素的个数是有限的; A 是动作空间; P 是转移概率; R 是立即奖赏函数; $C=\{c_i:S\times A\to R|i=1,\ldots,k\}$表示动作空间受约束的约束函数集合, 状态空间受约束的约束函数集合表示为 $C=\{\overline{c}_i:S\to R\,|\,i=1,...,k\}$. 在安全强化学习模型中, 智能体探索的目的是: 在有安全约束的情况下, 最大化长期累积回报奖赏. 安全强化学习模型示意图如图 2 所示.

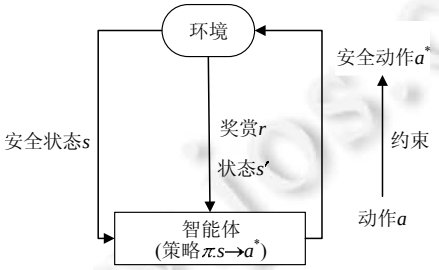

图 2 安全强化学习模型示意图

2.1 安全Sarsa(λ)模型

安全 Sarsa(λ)方法是一种基于受限马尔可夫决策过程的安全强化学习方法, 对状态空间添加约束的安全 Sarsa(λ)模型为

$$\left.\begin{aligned}&\arg\max_{a_{t+1}}Q(s_{t+1},a_{t+1})\leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))E_t(s_t,a_t)\\&\text{s.t. }\ \overline{c}_i(s_t)=\overline{C}_i,\ i=1,2,...,k'\\&\quad\ \overline{c}_i(s_t)\leqslant\overline{C}_i,\ i=k'+1,...,k\end{aligned}\right\}\tag{13}$$

对动作空间添加约束的安全 Sarsa(λ)模型为

$$\left.\begin{aligned}&\arg\max_{a_{t+1}} Q(s_{t+1},a_{t+1}) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))E_t(s_t,a_t)\\&\text{s.t. } c_i(s_t,a_t)=C_i,\ i=1,2,...,k'\\&\quad c_i(s_t,a_t)\leqslant C_i,\ i=k'+1,...,k\end{aligned}\right\} \tag{14}$$

指标集{1,2,…,k'}和{k'+1,k'+2,…,k}分别表示等式约束指标集和不等式约束指标集, s_t表示 t时刻智能体所处的状态, a_t表示 t 时刻处在状态 s_t下智能体选择的动作. 智能体在当前状态下, 只有选择了动作才能进入下一状态. 由于只对状态空间添加约束的安全 Sarsa(λ)模型不能体现这一动态过程, 而对动作空间添加约束集的安全 Sarsa(λ)模型可以很好地体现智能体的运行过程, 所以本文采用动作空间受约束的安全 Sarsa(λ)模型进行分析和求解.

在强化学习模型下, 通过添加约束条件, 可以将状态空间分为安全状态空间集合和非安全状态空间集合, 将动作空间分为安全动作空间集合和非安全动作空间集合, 可以确保在调度的最初时刻就解决安全问题并确保智能体的安全, 也可以在智能体进入某一状态之前通过约束条件判断该状态是否为安全状态, 只有安全状态智能体才可达. 故安全 Sarsa(λ)模型的可行域为

$$\bar{S}=\{s\mid \bar{c}_i(s)=C_i,i=1,...,k';\bar{c}_i(s)\leqslant C_i,i=k'+1,...,k\} \tag{15}$$

$$\bar{A}=\{a\mid c_i(s,a)=C_i,i=1,...,k';c_i(s,a)\leqslant C_i,i=k'+1,...,k\} \tag{16}$$

关于状态空间和动作空间的不等式约束 $\bar{c}_i(s_t)\leqslant C_i$, $c_i(s_t,a_t)\leqslant C_i$ 是标准形式, 如果对于某些状态满足不等式约束形式 $\bar{c}_i(s_t)\geqslant C_i$, $c_i(s_t,a_t)\geqslant C_i$, 只需要在不等式的两边同时乘以−1, 就可把不等式约束转化成标准形式. 对于不等式约束, 如果存在 $i_0\in\{k'+1,\ldots,k\}$, 使得 $\bar{c}_{i_0}(s_t)\leqslant \bar{C}_{i_0}, c_{i_0}(s_t,a_t)<C_{i_0}$, 在最优化问题模型中, 最优解满足条件$\lambda$=0, 或者 $c_i(x_t,u_t)-C_i=0$, $i=k'+1,\ldots,k$, 则第 i_0 个严格不等式约束在 s_t 处是无效约束, 故可以去掉该约束. 将有效约束和无效约束的概念推广到任意状态上, 可以得到下面集合:

$$\xi=\{1,2,\ldots,k'\},\ \zeta=\{k'+1,\ldots,k\},\ \zeta'=\{i\mid c_i(s,a)=C_i,i\in\zeta\} \tag{17}$$

根据公式(17), 在状态 s 处的有效约束指标集为$\xi\cup\zeta'$, 可以将不等式约束转化为等式约束, 安全 Sarsa(λ)模型修改为

$$\left.\begin{aligned}&\arg\max_{a_{t+1}} Q(s_{t+1},a_{t+1}) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))E_t(s_t,a_t)\\&\text{s.t. } c_i(s_t,a_t)=C_i,\ i\in\xi\cup\zeta'\end{aligned}\right\} \tag{18}$$

上述安全 Sarsa(λ)模型只包含了等式约束, 因为只有等式约束是有效约束, 去除的严格不等式约束是无效约束, 这样既降低了模型的复杂度, 也降低了模型求解的难度, 提高了模型的求解速度.

2.2 安全Sarsa(λ)模型求解

安全 Sarsa(λ)模型是在 Sarsa(λ)模型的基础上增加了多维约束, 故传统的强化学习问题的求解方法不再适用. 为了高效准确地求解安全 Sarsa(λ)模型, 本文提出用拉格朗日乘数法求解安全 Sarsa(λ)模型, 拉格朗日乘数法求解最优化问题的要求是: 目标函数和约束函数满足一阶连续可微条件[37], 目标函数在时间 t 连续的情况下是一阶连续可微的. 但是约束函数在构造的过程中不一定可以保证一阶连续可微, 但是可以通过对约束函数线性化实现约束函数的可微性. 由于智能体的下一状态是由当前状态和当前采取的动作决定的, 可得:

$$s'=\phi(s,a) \tag{19}$$

在模型求解过程中, 如果目标函数和约束函数是凸函数, 求解得到的结果是全局最优解. 根据式(18)可知, 目标函数是凸函数, 但是约束函数可能不是凸函数, 如果利用泰勒展开式[38]将约束函数进行线性近似, 根据线性函数是凸函数, 则线性化后得到的约束函数是凸函数. 此时求解安全 Sarsa(λ)模型, 可以得到全局最优解. 约束函数的一阶泰勒展开式为

$$c_i(s_t,a_t)-C_i\approx c_i(s_{t-1},a_{t-1})-C_i+\nabla c_i(s_t,a_t)a_t,i\in\xi\cup\zeta' \tag{20}$$

令 $\varphi_i(s_t,a_t)=c_i(s_{t-1},a_{t-1})-C_i+\nabla c_i(s_t,a_t)a_t$, $i\in\xi\cup\zeta'$, 模型中的约束函数可以表示为 $\varphi_i(s_t,a_t)=0$, $i\in\xi\cup\zeta'$的形式.

对约束函数进行线性近似后, 得到的安全 Sarsa(λ)模型为

$$\left.\begin{aligned}&\arg\max_{a_{t+1}} Q(s_{t+1},a_{t+1}) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))E_t(s_t,a_t)\\&\text{s.t. } \varphi_i(s_t,a_t)=0,\ i\in\xi\cup\zeta'\end{aligned}\right\} \tag{21}$$

安全 Sarsa(λ)模型可以采用拉格朗日乘数法进行求解, 首先将模型转化为

$$a^*=\arg\max_{a_{t+1}}\left\{Q(s_{t+1},a_{t+1}) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))E_t(s_t,a_t)-\sum_{i\in\xi\cup\zeta'}\lambda_i\varphi_i(s_t,a_t)\right\} \tag{22}$$

公式(22)是凸函数, 这样可以避免模型陷入局部最优解得不到最大长期累积回报值的问题. 最后采用梯度下降法求解公式(22).

本文采用线性化约束函数, 得到目标函数和约束函数都是凸函数的安全 Sarsa(λ)模型, 所以, 由拉格朗日乘数法求解得到的最优解是全局最优解. 命题 1 给出了动作空间受约束的安全强化学习模型的状态值函数和状态-动作值函数迭代收敛证明, 由于安全 Sarsa(λ)方法是同策略安全强化学习方法中的一种, 故安全 Sarsa(λ)模型的更新公式使用 TD 迭代更新后最终收敛到唯一的状态-动作最优值. 命题 2 是采用拉格朗日乘数法求解安全 Sarsa(λ)公式(22), 可以得到全局最优解的证明.

命题 1. 动作空间受约束的安全强化学习模型的状态值函数和状态-动作值函数迭代收敛, 分别收敛到唯一最优状态值和唯一最优状态-动作值.

证明: 将安全强化学习模型中的第 i 个约束函数 $c_i(s_t,a_t)\leqslant C_i$ 写为 $c_i(s_t,a_t)-C_i\leqslant 0$ 的形式, 并记为 $f_i(s_t,a_t)$. 状态值函数的贝尔曼等式为 $V_\pi(s)=E_\pi[G_t'\mid s_t=s]$, 对智能体可执行的动作空间添加约束后, 其迭代过程如下:

$$\left.\begin{aligned}V_\pi(s)&=E_\pi[G_t'\mid s_t=s]\\&=E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{k+t+1}-\sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\middle| s_t=s\right]\\&=E_\pi\left[R_{t+1}+\gamma G_{t+1}-\sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\middle| s_t=s\right]\\&=\sum_{a\in A}\pi(a\mid s)\sum_{s'\in S}P_{s\to s'}^a\left[R_{t+1}+\gamma E_\pi[G_{t+1}\mid s_{t+1}=s']-\sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\right]\\&=\sum_{a\in A}\pi(a\mid s)\sum_{s'\in S}P_{s\to s'}^a\left[R_{t+1}+\gamma V_\pi(s')-\sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\right]\end{aligned}\right\} \tag{23}$$

状态-动作值函数的贝尔曼等式为: $Q_\pi(s,a)=E_\pi[G_t'\mid s_t=s,a_t=a]$, 动作空间添加约束后, 状态-动作值函数的迭代过程如公式(24)所示.

由迭代公式(23)和公式(24)可知: 对智能体可执行的动作空间添加约束后不改变状态值和状态-动作值的收敛性, 安全强化学习模型的状态值通过不断迭代最终收敛于最优状态值, 状态-动作值通过不断迭代最终收敛于最优状态-动作值. □

命题 2. 假设 $\{a^*,\{\lambda_i^*\}_{i=1}^k\}$ 是安全 Sarsa(λ)模型(公式(21))的可行解, 其中, λ_i^* $\{i\in\xi\cup\zeta\}$是与第 i 个约束相关的最佳拉格朗日乘子, 证明安全 Sarsa(λ)模型的局部最优解为全局最优解.

$$\left.\begin{aligned}
Q_\pi(s,a) &= E_\pi[G'_t \mid s_t=s, a_t=a] \\
&= E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{k+t+1} - \sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\middle| s_t=s, a_t=a\right] \\
&= E_\pi\left[R_{t+1}+\gamma G_{t+1} - \sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\middle| s_t=s, a_t=a\right] \\
&= \sum_{s'\in S} P_{s\to s'}^a\left[R_{t+1}+\gamma E_\pi[G_{t+1}\mid s_{t+1}=s', a_{t+1}=a'] - \sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\right] \\
&= \sum_{s'\in S} P_{s\to s'}^a\left[R_{t+1}+\gamma V_\pi(s') - \sum_{i\in\xi\cup\zeta'}\lambda_i^t * f_i(s_t,a_t)\right]
\end{aligned}\right\} \tag{24}$$

证明: 设动作 a^*是公式(21)的局部最优解, 设 $Q(s_{t+1},a_{t+1})$, $\varphi_i(s_t,a_t)\{i\in\xi\cup\zeta'\}$在 a^*的邻域内一阶连续可微. 如果约束规范条件 $SFD(a^*,A)=LFD(a^*,A)$[39]成立, 则存在拉格朗日乘子 $\lambda_i^*\{i\in\xi\cup\zeta'\}$, 使得公式(25)–(28)成立:

$$\nabla Q(s_{t+1},a_{t+1}) = \sum_{i\in\xi\cup\zeta'}\lambda_i^*\nabla\varphi_i(s^*,a^*) \tag{25}$$

$$\varphi_i(s^*,a^*)=0,\ i\in\xi \tag{26}$$

$$\lambda_i^* \geqslant 0,\ i\in\zeta' \tag{27}$$

$$\lambda_i^*\varphi_i(s^*,a^*)=0,\ i\in\zeta' \tag{28}$$

由于 a^*是局部最优解, 故 a^*是可行解, 从而公式(24)成立. 设 $d\in SFD(a^*,A)$, 根据动作 a^*是局部最优解, 从几何最优性条件得到 $d^T\nabla Q(s^*,a^*)\leqslant 0$. 由约束规范条件 $d\in LFD(a^*,A)$, 因此, 方程组(29)–(31)无解:

$$d^T\nabla\varphi_i(s^*,a^*)=0,\ i\in\xi \tag{29}$$

$$d^T\nabla\varphi_i(s^*,a^*)\leqslant 0,\ i\in\zeta' \tag{30}$$

$$d^T\nabla Q(s^*,a^*)>0 \tag{31}$$

再利用 Farkas 引理[40]可得:

$$\nabla Q(s_{t+1},a_{t+1}) = \sum_{i\in\xi}\lambda_i^*\varphi_i(s^*,a^*) + \sum_{i\in\zeta'}\lambda_i^*\varphi_i(s^*,a^*) \tag{32}$$

最后, 显然有 $\lambda_i^*\geqslant 0$ 且 $\lambda_i^*\varphi_i(s^*,a^*)=0$, $i\in\xi$. 又因为安全 Sarsa(λ)模型为凸模型, 所以局部最优解 a^*为全局最优解. □

2.3 基于受限模型的安全Sarsa(λ)算法

Sarsa(λ)算法用来解决无模型的同策略的强化学习问题. 本文在 Sarsa(λ)算法的基础上, 提出了基于受限马尔可夫决策过程的安全 Sarsa(λ)算法. 该方法在最大化智能体长期回报值的基础上, 通过添加多维约束, 将智能体每一步执行的动作限制到安全动作集合上, 进而将智能体的可行状态集合限制到安全状态集合上, 在智能体探索的过程中保证了智能体的安全. 安全 Sarsa(λ)算法如算法 1 所示.

算法 1. 安全 Sarsa(λ)算法(algorithm of safe Sarsa(λ)).

1. 初始化: 状态-动作值 $Q(s,a)$, $s\in S$, $a\in A(s)$, 步长$\alpha\in(0,1]$, 拉格朗日乘子λ_i, $i=1,\ldots,k$, $k\in N^+$, 约束函数集合$\{c_i(s_t,a_t)=C_i, i\in\xi\cup\zeta'\}$, 并用泰勒展开式对约束函数进行线性化处理, 处理后的约束函数集合为$\{\varphi_i(s_t,a_t)=0, i\in\xi\cup\zeta'\}$.
2. **Repeat**(对每一个情节):
3. 初始化资格迹 $E(s,a)\leftarrow 0$, $s\in S$, $a\in A(s)$
4. 初始化状态空间 S 和动作空间 A 并选择初始安全动作 a
5. **Repeat**(对于情节中的每个时间步):
6. 执行动作 a, 观察回报值 r 和下一状态 s'

7. 求解最优动作: $a^* \leftarrow \begin{cases} \arg\max\limits_{a'} \left\{ Q(s',a') - \sum\limits_{i\in\xi\cup\zeta'} \lambda_i \varphi_i(s',a') \right\}, & \text{以 } \varepsilon \text{概率} \\ a,\ \varphi_i(s',a')=0,\ i\in\xi\cup\zeta',\ \forall a\in A(s), & \text{以} 1-\varepsilon \text{概率} \end{cases}$

8. $\delta \leftarrow r+\gamma Q(s',a^*)-Q(s,a)$

9. $E(s,a)\leftarrow E(s,a)+1$

10. **For each** $s\in S$, $a\in A(s)$:

11. $Q(s,a)\leftarrow Q(s,a)+\alpha\delta E(s,a)$

12. $E(s,a)\leftarrow \lambda\gamma E(s,a)$

13. $s\leftarrow s'$, $a\leftarrow a^*$

14. **Until** 智能体到达终止状态

在算法 1 中, 需要根据智能体所处的实验环境和可执行的动作集合对约束函数集合进行初始化. 之后, 应用泰勒展开式对每一个约束函数进行线性化, 保证约束函数是凸函数. 如果初始化后的约束函数是凸的并且是可微的, 则不需要对约束函数线性化. 算法中需要对拉格朗日乘子初始化, 在用梯度下降法求解模型后, 拉格朗日乘子会根据求解结果而发生改变, 因此, 拉格朗日乘子的初始化不影响问题的求解. 第 7 步采用 ε-greedy 方法求解在状态 s'时可以执行的一个动作 a', 算法以 ε的概率采用拉格朗日乘数法求解约束问题(21)得到在满足长期累积回报值最大的情况下的安全动作, 以 $1-\varepsilon$的概率随机选取满足约束条件$\{\varphi_i(s_t,a_t)=0, i\in\xi\cup\zeta'\}$的安全动作. 在第 13 步, 智能体进入下一状态 s', 并选择下一状态可以执行的安全动作 a^*. 安全 Sarsa(λ)算法在满足选取策略的情况下, 同时保证了智能体的安全.

2.4 基于受限模型的安全Sarsa算法

为进一步分析安全 Sarsa(λ)算法中, 资格迹λ对智能体在探索过程中安全问题的影响, 本文设计了安全 Sarsa 算法与安全 Sarsa(λ)算法进行对比. 由于本文提出的安全模型和求解方法可以应用在不同的强化学习方法中, 所以安全 Sarsa(λ)模型的设计方法及模型的求解方法可以应用在安全 Sarsa 模型的设计及模型的求解上.

在 Sarsa 模型的基础上, 对状态空间添加约束的安全 Sarsa 模型为

$$\left.\begin{aligned} &\arg\max_{a_t} Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)) \\ &\text{s.t. } \overline{c}_i(s_t)=\overline{C}_i,\ i=1,2,...,k' \\ &\quad \overline{c}_i(s_t)\leqslant \overline{C}_i,\ i=k'+1,...,k \end{aligned}\right\} \tag{33}$$

对动作空间添加约束的安全 Sarsa 模型为

$$\left.\begin{aligned} &\arg\max_{a_t} Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)) \\ &\text{s.t. } c_i(s_t,a_t)=C_i,\ i=1,2,...,k' \\ &\quad c_i(s_t,a_t)\leqslant C_i,\ i=k'+1,...,k \end{aligned}\right\} \tag{34}$$

为与对动作空间添加约束的安全 Sarsa(λ)模型进行对比, 本文选择对动作空间添加约束的安全 Sarsa 模型进行分析和求解. 安全 Sarsa 模型的可行域为

$$\overline{S}=\{s\,|\,\overline{c}_i(s)=C_i, i=1,...,k';\overline{c}_i(s)\leqslant C_i, i=k'+1,...,k\} \tag{35}$$

$$\overline{A}=\{a\,|\,c_i(s,a)=C_i, i=1,...,k';c_i(s,a)\leqslant C_i, i=k'+1,...,k\} \tag{36}$$

将不等式约束转化为等式约束后, 安全 Sarsa 模型如下所示:

$$\left.\begin{aligned} &\arg\max_{a_t} Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)) \\ &\text{s.t. } c_i(s_t,a_t)=C_i,\ i\in\xi\cup\zeta' \end{aligned}\right\} \tag{37}$$

应用泰勒展开式对约束函数进行线性近似后, 得到的安全 Sarsa 模型为

$$\left.\begin{aligned}&\arg\max_{a_t} Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))\\&\text{s.t. } \varphi_i(s_t,a_t)=0,\ i\in\xi\cup\zeta'\end{aligned}\right\} \tag{38}$$

根据拉格朗日乘数法求解上述模型, 即求解如下公式:

$$a^*=\arg\max_{a_t}\left\{Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha(R_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t))-\sum_{i\in\xi\cup\zeta'}\lambda_i\varphi_i(s_t,a_t)\right\} \tag{39}$$

为了避免模型陷入局部最优解而得不到最大长期累积回报值, 本文采用线性化约束函数, 得到目标函数和约束函数都是凸函数的安全 Sarsa 模型. 所以, 由拉格朗日乘数法求解得到的最优解是全局最优解, 证明过程类似于命题 2, 值函数迭代收敛性证明同命题 1.

根据安全 Sarsa 模型的设计和模型的求解过程, 安全 Sarsa 算法如算法 2 所示.

算法 2. 安全 Sarsa 算法(algorithm of safe Sarsa).

1. 初始化: 状态-动作值 $Q(s,a)$, $s\in S$, $a\in A(s)$, 步长$\alpha\in(0,1]$, 拉格朗日乘子λ_i, $i=1,\ldots,k$, $k\in N^+$, 约束函数集合$\{c_i(s_t,a_t)=C_i,i\in\xi\cup\zeta\}$, 并用泰勒展开式对约束函数进行线性化处理, 处理后的约束函数集合为$\{\varphi_i(s_t,a_t)=0,i\in\xi\cup\zeta\}$.
2. **Repeat**(对每一个情节):
3. 初始化状态空间 S 和动作空间 A 并选择初始安全动作 a
4. **Repeat**(对于情节中的每个时间步):
5. 执行动作 a,观察回报值 r 和下一状态 s'
6. 求解在状态 s'时智能体可执行的最优动作:

$$a^*\leftarrow\begin{cases}\arg\max_{a'}\left\{Q(s',a')-\sum_{i\in\xi\cup\zeta'}\lambda_i\varphi_i(s',a')\right\}, & \text{以 }\varepsilon\text{概率}\\ a,\ \varphi_i(s',a')=0,\ i\in\xi\cup\zeta',\ \forall a\in A(s), & \text{以 }1-\varepsilon\text{概率}\end{cases}$$

7. $Q(s,a)\leftarrow Q(s,a)+\alpha[r+\gamma Q(s',a^*)-Q(s,a)]$
8. $s\leftarrow s'$, $a\leftarrow a^*$
9. **Until** 智能体到达终止状态

算法2中, 首先对拉格朗日乘子初始化, 方便问题的求解, 并根据智能体可执行的动作空间集及具体的实验环境给出为保证智能体的安全所需要满足的约束函数集合, 如果约束函数非凸, 则用泰勒展开式对约束函数进行线性化处理. 第 6 步采用ε-greedy 方法求解状态 s'下需要执行的动作, 以ε的概率采用拉格朗日乘数法求解约束问题, 得到在满足长期累积回报值最大的情况下的安全动作, 以 $1-\varepsilon$的概率随机选择一个满足约束条件集的安全动作. 在第 8 步智能体进入下一状态 s', 并选择下一状态的安全动作 a^*. 安全 Sarsa 算法在满足求解得到最优策略的情况下, 同时保证了智能体的安全.

3 实验结果及分析

基于受限马尔可夫决策过程的安全 Sarsa(λ)算法和安全 Sarsa 算法可以用于智能体在受限情况下的探索学习, 以得到最大的长期累积回报值. 本文提出的安全 Sarsa(λ)算法和安全 Sarsa 算法主要用于解决智能体的安全问题, 避免智能体在运行过程中陷入危险状态. 在下面的实验中, 危险状态指当智能体进入该状态时, 会导致任务失败的状态.

悬崖行走实验和带陷阱的迷宫实验中, 为保证智能体的安全, 智能体在运行过程中不能进入危险状态. 智能体所处的当前状态到危险状态的距离采用曼哈顿距离[41], 要保证智能体的安全, 就要使智能体采取某一动作到达的下一状态与危险状态的距离大于等于 1, 当前状态与危险状态的距离定义为安全距离 d, 安全约束函数为

$$d(s_{t+1},\chi)>1 \tag{40}$$

其中, χ是由所有的危险状态组成的集合, s_{t+1}是智能体即将进入的下一个状态. 由状态转移函数 $s_{t+1}=\phi(s_t,a_t)$可以得到公式(40)的等价形式, 如公式(41)所示:

$$d(\phi(s_t,a_t),\chi)>1 \tag{41}$$

3.1 悬崖行走实验

悬崖行走(cliff walking)实验[42]中, 智能体需要在不跌落悬崖的前提下, 由起点到终点寻找一条安全的最短路径, 如图 3 所示, 其中, S 是起点, G 是终点. 灰色的部分是危险状态悬崖, 如果智能体走到悬崖, 则得到 –0.5 的奖赏, 表示对智能体进入危险状态的惩罚. 除了危险状态和终止状态, 为避免智能体在网格中随意漫步, 智能体每次进入一个新的状态都会得到–0.01 的奖赏, 这就避免了智能体在网格中随意行走, 保证智能体可以在最短的时间内寻找最短的路径进入终止状态. 在图 3 中: 蓝线代表的路径是最优路径, 智能体可以在保证安全的前提下, 在最短的时间内由起点走到终点; 黄线代表的路径仅仅是安全路径, 也就是智能体在由起点到终点的过程中所经历的状态离悬崖越远, 智能体越安全, 但是这往往不是最优路径.

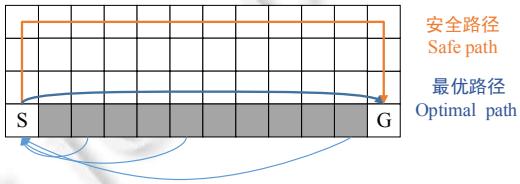

图 3 悬崖行走示意图

实验对比了 3 种传统强化学习方法 Sarsa, Sarsa(λ), Q-Learning 和 3 种安全强化学习方法安全 Sarsa(λ)、安全 Sarsa 和 RCPO 在有危险的环境中智能体的学习性能. 图 4 是实验结果对比图, 图中横坐标为训练情节数, 纵坐标为平均每情节的长期累积回报值. 在参数设置部分, Sarsa、Sarsa(λ)、Q-Learning、安全 Sarsa(λ)和安全 Sarsa 的步长α统一设置为 0.5, 折扣因子γ设置为 1, λ取值为 0.9, 策略参数ε设置为 0.1, ε-greedy 方法用于智能体对随机动作的选择. RCPO 算法将约束作为一个惩罚项整合到回报值函数中, 通过惩罚项的指导求解满足约束的策略[25]. RCPO 算法的参数保留了原始设置, 学习率设置为 0.01, 折扣因子γ设置为 0.99, λ初始化为 0. 实验一共独立运行 50 次, 每一次独立运行的情节数为 500.

悬崖行走实验使用安全 Sarsa(λ)算法和安全 Sarsa 算法得到的每情节平均长期累积回报值, 较 Sarsa, Sarsa(λ), Q-Learning 和 RCPO 算法都有显著的提高. 由于 Q-Learning 算法生成行为使用的策略和更新值函数使用的策略不是同一个策略, 所以 Q-Learning 是一种典型的异策略算法, 而 Sarsa 和 Sarsa(λ)是一种同策略算法. 基于 Q-Learning, Sarsa(λ)和 Sarsa 算法, 智能体在探索过程中不能避免进入危险状态, 每次进入危险状态, 智能体都得到–0.5 的惩罚奖赏, 所以 Q-Learning, Sarsa(λ)和 Sarsa 算法计算得到的平均每情节的长期累积回报值并不稳定, 智能体极易进入危险状态, 具有较大的波动性, 如图 4 中阴影部分所示, 其中, Q-Learning 算法计算得到的平均每情节的长期累积回报值最不稳定, 并且学习效果最不好. RCPO 算法在悬崖行走实验中的实验效果较差, 收敛速度慢, 实验运行到 450 情节左右才收敛. 安全 Sarsa(λ)算法和安全 Sarsa 算法能得到更好的平均每情节的长期累积回报值, 这是因为通过添加约束保证了智能体的安全, 智能体不会进入危险状态而造成不必要的损失, 故而安全算法收敛速度较快, 学习效果较好. 由图 4 可得, 安全 Sarsa(λ)算法和安全 Sarsa 算法的效果相似, 这与 Sarsa(λ)算法和 Sarsa 算法实验结果的相似性是一致的, 也说明资格迹λ没有影响安全算法的实验效果.

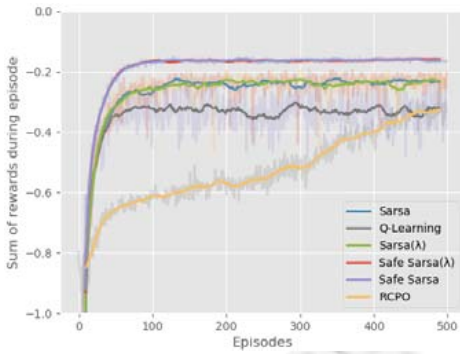

图 4　6 种方法在悬崖漫步实验中的效果图

3.2 带陷阱的迷宫实验

带陷阱的迷宫实验中, 智能体探索的目的是在不进入陷阱的前提下, 寻找一条由起点到终点的安全最短路径. 迷宫实验是一个 6×6 的网格, 如图 5 所示, 其中, 蓝色格代表陷阱, G 代表目标终点, S 代表起始位置. 若智能体进入蓝色格, 则代表智能体遇到危险, 智能体会得到–10 的奖赏, 智能体到达 G 代表智能体到达目标终点, 整个情节结束. 为避免智能体在格子中随意漫步, 智能体每到达一个新的非危险状态, 都会得到–1 的奖赏, 这会使智能体在最短的时间内由起点到达终点, 即寻找一条由起点到终点最短的安全路径.

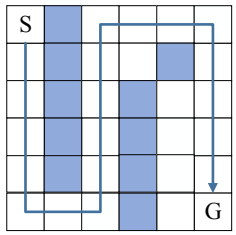

图 5　带陷阱的迷宫示意图

实验对比了 Sarsa 算法、Sarsa(λ)算法、Q-Learning 算法、安全 Sarsa(λ)算法、安全 Sarsa 算法和 RCPO 算法在有陷阱的迷宫实验环境中, 智能体的学习性能. Sarsa 算法、Sarsa(λ)算法、Q-Learning 算法、安全 Sarsa(λ)算法和安全 Sarsa 算法的实验参数设置相同, 步长 α统一设置为 0.1, 折扣因子 γ统一设置为 0.9, λ取值为 0.9, ε-greedy 策略的参数 ε设置为 0.8. RCPO 算法的学习率设置为 0.01, 折扣因子 γ设置为 0.99, λ初始化为 0. 实验的情节数 episode 等于 500, 实验一共独立运行 50 次. 图 6 显示了 Sarsa、Sarsa(λ)、Q-Learning、安全 Sarsa(λ)、安全 Sarsa 和 RCPO 这 6 种强化学习算法在迷宫实验中每情节的长期累积回报值.

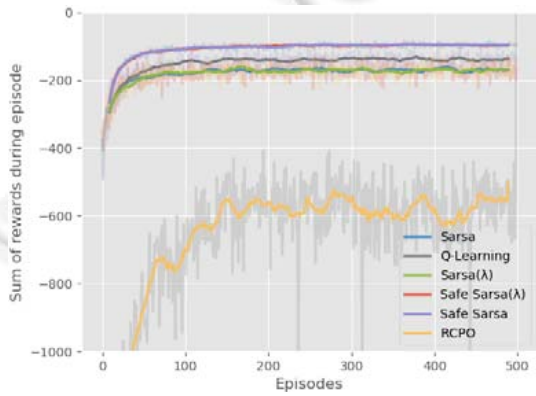

图 6　6 种方法在带陷阱的迷宫实验中的效果图

在带陷阱的迷宫实验中, 由安全 Sarsa(λ)算法和安全 Sarsa 算法计算所得的平均每情节的长期累积回报值较 Sarsa, Sarsa(λ), Q-Learning 和 RCPO 算法计算所得的有显著提高. 这是因为在安全 Sarsa(λ)算法和安全 Sarsa 算法下, 智能体在学习过程中不会掉入陷阱得到较小的负回报值, 而 Sarsa, Sarsa(λ)和 Q-Learning 算法下智能体会因试错式的学习方式反复地进入陷阱, 所以计算所得的平均每情节的长期累积回报值较低, 且不稳定. 安全 Sarsa(λ)算法和安全 Sarsa 算法可以保证智能体在探索过程中的安全, 避免了智能体进入危险状态带来的损失. RCPO 算法的实验效果非常差, 智能体虽然也会通过学习使平均每情节长期累积回报值收敛, 但是花费时间长, 且最终收敛效果差. 图 6 表明, 安全 Sarsa(λ)算法和安全 Sarsa 算法的表现效果相似, 资格迹λ不会影响安全算法的实验效果. 实验结果表明: 拉格朗日乘数法求解过程不受资格迹的影响, 可以与强化学习算法结合解决智能体的安全问题.

3.3 Mountain Car实验

Mountain Car 实验[42]中, 智能体是一辆动力不足的汽车, 汽车探索的目的是在最短的时间内由坡底到达目标 Goal, 如图 7 所示. 汽车在陡坡上向上行驶时, 由于汽车动力不足, 且重力比汽车的发动机更强, 导致汽车无法在陡坡上加速, 只能减速行驶. 为使汽车到达目的地, 唯一的解决办法是汽车先向左上方加速前进, 提升自己的高度, 在此过程中, 汽车可以积累足够多的惯性, 并在发动机的作用下通过右面的陡坡, 到达目的地. Mountain Car 是强化学习问题中情节式连续空间问题.

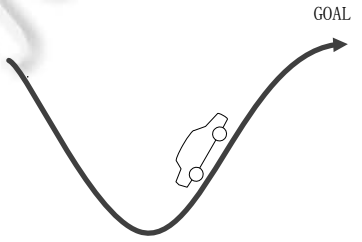

图 7 Mountain Car 示意图

在 Mountain Car 实验中, 汽车有加足马力前进、加足马力后退和原地不动这 3 种动作, 为避免汽车在探索过程中随意漫步, 汽车每执行一个动作进入下一状态都会得到–1 的奖赏, 直到汽车通过目标, 整个情节结束. 记汽车在探索过程中 t 时刻的位置是 l_t, 此时汽车的速度是 v_t, 汽车的位置和速度的更新公式如下:

$$l_{t+1}=bound[l_t+v_{t+1}] \tag{42}$$

$$v_{t+1}=bound[v_t+0.001\alpha-0.0025\cos(3l_t)] \tag{43}$$

上述公式中, 位置和速度的取值范围分别是 $l_{t+1}\in[-1.2,0.5]$, $v_{t+1}\in[-0.07,0.07]$. 如果汽车到达位置的最左端, 即 $l_{t+1}=-1.2$, 汽车的速度会置为 0, 此时汽车不会再向左上方移动, 开始沿坡下滑. 汽车到达位置的最右端, 即汽车到达目标, 整个情节终止. 汽车开始运动时速度为 0, 起始位置为区间[–0.6,–0.4]中的任意位置. 为了用二进制特征表示位置和速度两个连续变量, 本文采用有 8 个 Tilings 的 grid-tilings 方法进行转换, 在每个维度上, 一个 Tiling 覆盖 1/8 的范围, 并将特征向量用参数进行线性组合, 代替原来的状态动作值函数.

在实际应用中, 由于目标位置是坡顶, 汽车在到达目标位置时如果速度太大, 会导致汽车被平抛出去等危险事件发生. 当汽车在坡顶时, 对汽车进行受力分析可得, 汽车的最大速度满足公式(44), 即汽车到达坡顶时速度需要满足公式(45), 否则汽车会进入危险状态, 造成损害:

$$G-F_N=mv^2/R \tag{44}$$

$$v\leqslant\sqrt{(G-F_N)R/m} \tag{45}$$

其中, G 是汽车的重力, R 是坡顶所在圆弧的半径, m 是汽车的质量, F_N 是汽车受坡顶的支持力. 根据上述分析, 为保证汽车在整个探索过程中的安全, 需要对汽车的速度进行限制, 即汽车在坡顶的速度需要满足公式(45), 否则汽车进入危险状态, 得到–20 的回报值.

Mountain Car 实验效果图如图 8 所示, 由图可知, 步长的选取会影响实验效果: 步长α取值越大, 智能体的学习效果越好. 安全 Sarsa(λ)-0.1/8 算法、安全 Sarsa(λ)-0.2/8 算法和安全 Sarsa(λ)-0.5/8 算法较 Sarsa(λ)-0.1/8 算法、Sarsa(λ)-0.2/8 算法和 Sarsa(λ)-0.5/8 算法的实验效果好, 但是收敛后的长期累积回报值的改善不是很大. 这是因为智能体在探索的整个过程中, 只有到达坡顶并且速度不满足公式(45)才会进入危险状态, 得到–20 的回报值. 对于安全 Sarsa(λ)-0.1/8 算法、安全 Sarsa(λ)-0.2/8 算法和安全 Sarsa(λ)-0.5/8 算法, 由于对智能体到达坡顶时的速度进行限制, 智能体不会进入危险状态, 保证了智能体的安全. 根据实验要求, 智能体到达坡顶时的速度不仅取决于智能体在向右上坡时的重力和发动机, 还受到智能体向左边的陡坡行走的距离也会影响汽车到达目标坡顶时的速度. 在限制条件下, 汽车到达坡顶的速度不会太大, 所以汽车不会向左上方行走太远的距离,所以汽车的探索步数相对较少, 探索过程收敛较快.

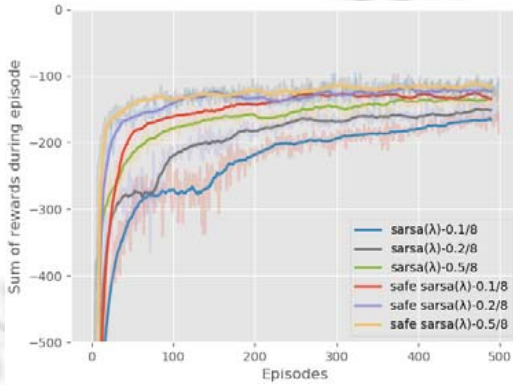

图 8 不同方法在 Mountain Car 实验中的效果图

3.4 Frozen Mars Rover实验

在悬崖行走实验和带陷阱的迷宫实验中, RCPO 算法的实验效果较差. 为进一步比较分析 RCPO 算法和本文提出的安全 Sarsa(λ)算法及安全 Sarsa 算法的性能, 我们用安全 Sarsa(λ)、安全 Sarsa、RCPO 算法以及 Sarsa, Sarsa(λ), Q-Learning 实现了 Frozen Mars Rover 实验[25], 该实验是参考文献[25]中的第 1 个实验, 用来分析安全算法 RCPO 的性能, 本文中的 Frozen Mars Rover 实验采用了与参考文献[25]中相同的实验环境以及参数设置. 实验环境如图 9 所示, Frozen Mars Rover 实验是一个 8×8 的格子世界, 图中棕色格是危险状态, 危险状态分布比较离散, 这与危险状态分布较为集中的悬崖行走实验和带陷阱的迷宫实验有所不同. 图 9 中状态 S 是起点, 状态 G 是目标终点.

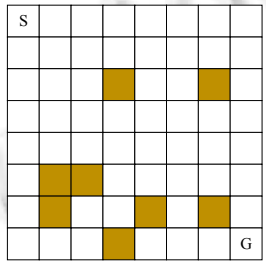

图 9 Frozen Mars Rover 示意图

在 Frozen Mars Rover 实验中, 智能体进入棕色的危险状态会得到–0.5 的奖赏值, 同时结束本情节并回到起始状态, 负奖赏值表示对智能体的惩罚. 智能体到达目标状态 G 会得到+1 的奖赏值, 同时终止本情节回到起始状态. 为避免智能体在环境中随意行走, 智能体进入任意非危险状态和非终止状态都会得到一个–0.01 的奖赏值, 该奖赏值使智能体更快地探索目标状态 G. 在对实验参数的设置中, RCPO 算法的学习率设置为 0.01,

折扣因子γ设置为 0.99, λ初始化为 0, 安全 Sarsa(λ)的学习率设置为 0.01, 折扣因子γ设置为 0.9, λ取值为 0.9, 策略参数ε设置为 0.9, ε-greedy 方法用于智能体对随机动作的选择, 安全 Sarsa 算法的学习率设置为 0.01, 折扣因子γ设置为 0.9, 策略参数ε设置为 0.9.

Frozen Mars Rover 实验效果图如图 10 所示, 该实验中, 安全 Sarsa(λ)算法和安全 Sarsa 算法可以指导智能体避开危险状态保证自身安全, 与其他算法相比较, 安全 Sarsa(λ)算法和安全 Sarsa 算法收敛速度快, 收敛效果好. 在该实验中, 安全 Sarsa 算法优于安全 Sarsa(λ)算法. RCPO 算法虽然在一定程度上也可以指导智能体的行为选择, 保证智能体的安全, 但是与安全 Sarsa(λ)算法和安全 Sarsa 算法相比效果不是很好, 并且 RCPO 算法在 400 情节左右才开始收敛, 收敛速度较慢. Sarsa, Sarsa(λ)和 Q-Learning 算法难以保证智能体的安全, 在对随机动作的选择过程中, 智能体极有可能选择危险动作, 进入危险状态. 实验表明, 安全 Sarsa(λ)算法和安全 Sarsa 算法可以较好地保证智能体的安全.

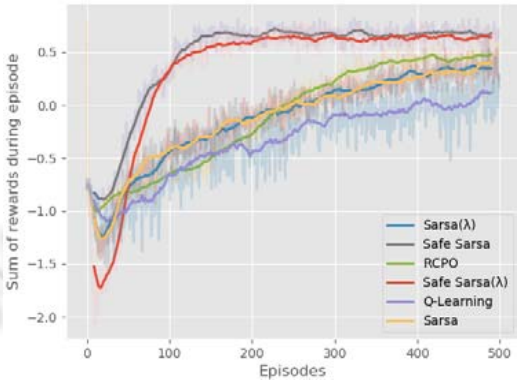

图 10 6 种方法在 Frozen Mars Rover 实验中的效果图

4 总 结

随着强化学习在工业领域中应用的不断推广, 智能体的安全问题越来越得到重视. 针对智能体的安全问题, 本文提出了基于受限马尔可夫决策过程的安全 Sarsa(λ)算法和安全 Sarsa 算法, 并分别对状态空间和动作空间为离散空间的悬崖行走、带陷阱的迷宫和 Frozen Mars Rover 实验以及对状态空间和动作空间为连续空间的 Mountain Car 进行实验, 实验结果表明: 安全算法可以将智能体的可行动作空间和状态空间限制到安全动作空间和安全状态空间上, 保证了智能体的安全. 安全 Sarsa(λ)算法和安全 Sarsa 算法的实验对比结果说明, 资格迹λ对本文提出的安全算法没有显著影响. 受限马尔可夫决策过程可以和不同的强化学习算法结合, 解决智能体在运行过程中的安全问题.

本文提出的安全 Sarsa(λ)算法和安全 Sarsa 算法可以保证智能体在探索过程中的安全, 安全 Sarsa(λ)算法和安全 Sarsa 算法是基于受限马尔可夫决策过程的, 所以该算法也可以推广和应用到受限的模型中, 例如资源有限或要求成本最小的模型、多目标模型以及对智能体速度有限制的模型中.

安全 Sarsa(λ)算法和安全 Sarsa 算法可以进一步推广到受限问题的建模和求解上. 因此, 下一步的研究工作可以尝试通过应用安全 Sarsa(λ)算法和安全 Sarsa 算法解决智能体的受限问题.

References:

[1] Liu Q, Zhai JW, Zhang ZZ, *et al*. A survey on deep reinforcement learning. Chinese Journal of Computer, 2018, 41(1): 1−27 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.00001]

[2] Xu X, Zuo L, Huang Z. Reinforcement learning algorithms with function approximation: Recent advances and applications. Information Sciences, 2014, 261(5): 1−31. [doi: 10.1016/j.ins.2013.08.037]

[3] Sallab A, Abdou M, Perot E, *et al*. Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017, 2017(19): 70−76. [doi: 10.2352/ISSN.2470-1173.2017.19.AVM-023]

[4] Madani K, Hooshyar M. A game theory—Reinforcement learning (GT-RL) method to develop optimal operation policies for multi-operator reservoir systems. Journal of Hydrology, 2014, 519: 732−742. [doi: 10.1016/j.jhydrol.2014.07.061]

[5] Mahmud M, Kaiser MS, Hussain A, *et al*. Applications of deep learning and reinforcement learning to biological data. IEEE Trans. on Neural Networks & Learning Systems, 2018, 29(6): 2063−2079. [doi: 10.1109/TNNLS.2018.2790388]

[6] Pagano A, Giordano R, Portoghese I, *et al*. A bayesian vulnerability assessment tool for drinking water mains under extreme events. Natural Hazards, 2014, 74(3): 2193−2227. [doi: 10.1007/s11069-014-1302-5]

[7] García, J, Fernández F. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 2015, 16: 1437−1480. [doi: 10.5555/2789272.2886795]

[8] Pecka M, Svoboda T. Safe exploration techniques for reinforcement learning—An overview. Modelling and Simulation for Autonomous Systems, 2014, 8906: 357−375. [doi: 10.1007/978-3-319-13823-7_31]

[9] Zhang H, Wang J, Zhou Z, *et al*. Learning to design games: strategic environments in reinforcement learning. In: Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence (IJCAI-18). 2017. 3068−3074. [doi: 10.24963/ijcai.2018/ 426]

[10] Ghavamzadeh M, Mannor S, Pineau J, *et al*. Bayesian reinforcement learning: A survey. Foundations and Trends in Machine Learning, 2015, 8(5−6): 359−483. [doi: 10.1561/2200000049]

[11] Tamar A, Xu H, Mannor S. Scaling up robust MDPs by reinforcement learning. In: Proc. of the 31st Int'l Conf. on Machine Learning. 2014. 181−189. [doi: 10.5555/3044805.3044913]

[12] Singh S, Lacotte J, Majumdar A, *et al*. Risk-sensitive inverse reinforcement learning via semi- and non-parametric methods. The Int'l Journal of Robotics Research, 2018, 37: 1−45. [doi: 10.1177/0278364918772017]

[13] Horie N, Matsui T, Moriyama K, *et al*. Multi-objective safe reinforcement learning: The relationship between multi-objective reinforcement learning and safe reinforcement learning. Artificial Life and Robotics, 2019, 24: 1−9. [doi: 10.1007/s10015-019-00523-3]

[14] Tommaso M, Kampen EJV, Visser CD, *et al*. Safe exploration algorithms for reinforcement learning controllers. IEEE Trans. on Neural Networks and Learning Systems, 2018, 29(4): 1069−1081. [doi: 10.1109/TNNLS.2017.2654539]

[15] Driessens K, Dzeroski S. Integrating guidance into relational reinforcement learning. Machine Learning, 2004, 57(3): 271−304. [doi: 10.1023/B:MACH.0000039779.47329.3a]

[16] Abbeel P, Coates A, Ng AY. Autonomous helicopter aerobatics through apprenticeship learning. Int'l Journal of Robotic Research, 2010, 29(13): 1608−1639. [doi: 10.1177/0278364910371999]

[17] Calinon S. A tutorial on task-parameterized movement learning and retrieval. Intelligent Service Robotics, 2016, 9(1): 1−29. [doi: 10.1007/s11370-015-0187-9]

[18] Chow Y, Tamar A, Mannor S, *et al*. Risk-sensitive and robust decision-making: A cvar optimization approach. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. 2015. 1522−1530. [doi: 10.5555/2969239.2969409]

[19] Bušić A, Meyn S. Action-Constrained Markov decision processes with kullback-leibler cost. arXiv: 1807.10244, 2018.

[20] Alshiekh M, Bloem R, Ehlers R, *et al*. Safe reinforcement learning via shielding. In: Proc. of the AAAI Conf. on Artificial Intelligence. 2018. 2669−2678.

[21] Fulton N, Platzer A. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. 2018. 6485−6492.

[22] Chow Y, Nachum O, Duenez-Guzman E, *et al*. A Lyapunov-based approach to safe reinforcement learning. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. 2018. 8103−8112.

[23] Cheng R, Orosz G, Murray RM, *et al*. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In: Proc. of the AAAI Conf. on Artificial Intelligence. 2019. 1−9. [doi: 10.1609/aaai.v33i01.33013387]

[24] Achiam J, Held D, Tamar A, *et al*. Constrained policy optimization. In: Proc. of the 34th Int'l Conf. on Machine Learning. 2017. 1−18.

[25] Tessler C, Mankowitz DJ, Mannor S. Reward constrained policy optimization. In: Proc. of the Int'l Conf. on Learning Representations. 2019. 1−15.

[26] Miryoosefi S, Brantley K, Daumé III H, *et al*. Reinforcement learning with convex constraints. In: Proc. of the 33rd Conf. on Neural Information Processing Systems. 2019. 1−10.

[27] Hasanbeig M, Abate A, Kroening D. Cautious reinforcement learning with logical constraints. In: Proc. of the 19th Int'l Conf. on Autonomous Agents and Multiagent Systems. 2020. 483−491. [doi: 10.5555/3398761.3398821]

[28] Altman, E. Constrained Markov Decision Processes. Chapman and Hall/CRC, 1999. [doi: 10.1007/BF02204825]

[29] Sutton RS, Barto AG. Reinforcement learning. A Bradford Book, 1998, 15(7): 665−685. [doi: 10.1007/978-3-642-27645-3]

[30] Zhu F, Liu Q, Fu QM, *et al*. A policy search and transfer approach in the non-stationary environment. Acta Electronica Sinica, 2017, 45(2): 3−12 (in Chinese with English abstract). [doi: 10.3969/j.issn.0372-2112.2017.02.001]

[31] Zhu F, Wu W, Liu Q, *et al*. A deep Q-network method based on upper confidence bound experience sampling. Journal of Computer Research and Development, 2018, 55(8): 100−111 (in Chinese with English abstract) [doi: 10.7544/issn1000-1239.2018. 20180148]

[32] Geist M, Scherrer B. Off-policy learning with eligibility traces: A survey. Journal of Machine Learning Research, 2013, 15(1): 289−333. [doi: 10.1007/s10883-013-9206-3]

[33] Golden RM. Adaptive learning algorithm convergence in passive and reactive environments. Neural Computation, 2018, 30(10): 2805−2832. [doi: 10.1162/neco_a_01117]

[34] Luo B, Liu D, Huang T, *et al*. Model-free optimal tracking control via critic-only Q-learning. IEEE Trans. on Neural Networks and Learning Systems, 2016, 27(10): 2134−2144. [doi: 10.1109/TNNLS.2016.2585520]

[35] Martyna J. Power allocation in cognitive radio with distributed antenna system. Lecture Notes in Computer Science, 2017, 10531: 745−754. [doi: 10.1007/978-3-319-67380-6_70]

[36] Farina F, Garulli A, Giannitrapani A, *et al*. Asynchronous distributed method of multipliers for constrained nonconvex optimization. In: Proc. of the 2018 European Control Conf. (ECC 2018). 2018. 103:243−253. [doi: 10.23919/ECC.2018.8550098]

[37] De Borbón ML, Ochoa P. A capacity-based condition for existence of solutions to fractional elliptic equations with first-order terms and measures. Potential Analysis, 2020, 1−22. [doi: 10.1007/s11118-020-09873-1]

[38] Abdi H. Linear algebra for neural networks. Int'l Encyclopedia of the Social & Behavioral Sciences, 2001, 8864−8868. [doi: 10.1016/B0-08-043076-7/00609-4]

[39] Klamroth K, Tind J. Constrained optimization using multiple objective programming. Journal of Global Optimization, 2007, 37(3): 325−355. [doi: 10.1007/s10898-006-9052-x]

[40] Cánovas JM, Dinh N, Long DH, *et al*. An approach to calmness of linear inequality systems from Farkas lemma. Optimization Letters, 2019, 13(2): 295−307. [doi: 10.1007/s11590-018-01380-y]

[41] Blackburn SR, Homberger C, Winkler P. The minimum Manhattan distance and minimum jump of permutations. Journal of Combinatorial Theory (Series A), 2019, 161: 364−386. [doi: 10.1016/j.jcta.2018.09.002]

[42] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. MIT Press Cambridge, 2018.

附中文参考文献:

[1] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述. 计算机学报, 2018, 41(1): 1−27. [doi: 10.11897/SP.J. 1016.2018.00001]

[30] 朱斐, 刘全, 傅启明, 等. 一种不稳定环境下的策略搜索及迁移方法. 电子学报, 2017, 45(2): 3−12. [doi: 10.3969/j.issn.0372-2112.2017.02.001]

[31] 朱斐, 吴文, 刘全, 等. 一种最大置信上界经验采样的深度 Q 网络方法. 计算机研究与发展, 2018, 55(8): 100−111. [doi: 10.7544/issn1000-1239.2018.20180148]

朱斐(1978－), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为安全强化学习, 深度强化学习, 医学信息学.

凌兴宏(1968－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为机器学习, 语义 Web, 知识管理, 企业信息化.

葛洋洋(1995－), 女, 硕士生, 主要研究领域为安全强化学习.

刘全(1969－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为深度学习, 强化学习, 统计人工智能.