

多智能体强化学习研究

Study on Reinforcement Learning for Multi Agents

北京理工大学机器人研究中心 童亮 龚建伟 熊光明 陆际联等

Robotics Research Center, Beijing Institute of Technology.

L. Tong, J.W. Gong, G.M. Xiong, J.L. Lu

转载此文请署名作者 并标明来自龚建伟技术主页 www.gjwtech.com

此文工作已在学术期刊上正式发表

多智能体强化学习研究.....	1
Study on Reinforcement Learning for Multi Agents	1
1 研究多智能体系统的必要性.....	2
2 多智能体学习方法研究.....	3
2.1 多智能体学习的框架.....	3
2.2 双矩阵决策和马尔可夫决策过程.....	5
2.3 随机决策.....	7
3 多智能体系统中的强化学习.....	8
3.1 智能体强化学习方法分类.....	9
3.2 Hu 和 Wellman 算法	11
4 基于 SLA 进行行动预测的多智能体强化学习算法.....	12
4.1 基于 SLA 进行行动预测的多智能体强化学习算法.....	13
4.2 多机器人推箱子问题.....	15
4.3 试验及结果比较.....	16
5 小结.....	17

1 研究多智能体系统的必要性

随着物理机器人和软件智能体的不断普及，对于多智能体的需求和应用，如足球机器人、搜索和营救、自动驾驶以及电子商务与信息智能体，变得越来越普遍。

对于单一智能体在静态环境中行动的学习，研究人员已经进行了大量的研究工作，而且在这些工作中应用智能体技术有以下几个优点：应用学习方法由于不需要精确的环境模型及对这个模型的最优化处理，从而大大简化了智能体的编程问题。学习也使得机器人可以适应未知和变化的环境。在多智能体环境，智能体的学习变得更加重要也更加困难。

在多智能体领域，智能体必须与其它智能体交互，它们可能具有不同的目标、假设、算法和协议。智能体为了处理这种环境，它们必须有适应其它智能体的能力。因为其它智能体也具有适应能力，这一点违背了传统行为学习的基本静态假设，使得学习的问题变得比较困难。因为其它智能体也在利用与环境交互的经验提高它们的操作水平，智能体依赖于其它智能体的策略使得对期望策略的定义也变得非常困难。本章主要介绍在存在其它智能体的复杂环境中智能体的评价学习方法。

事实上，由于存在各种限制条件，智能体并不是常常可以采取最优行动。它们可能有物理限制（如执行器坏掉或部分感知），使得智能体不可能执行特定的行动；也可能在学习任务中采用近似或抽象的概念，因此为了学习速度而牺牲最优。智能体也可能什么都学不到。在巨大而复杂的环境中，限制不可避免，特别是存在其它的智能体的环境中，使得智能体的行为可能没有理性。在实际应用的多智能体系统中必须强调包括智能体本身和其它智能体带来的限制。有效学习的智能体必须有能力弥补自身和它们的同伴或对手带来的限制。

对于学习，是指智能体通过与环境的不断交互得到的经验中提高其达到目标的能力或未来的累积回报过程。学习发生在智能体与环境的交互过程中：从环境中获得感知和回报并通过行动来改变环境。学习的复杂性来源于在环境中执行行动的其它智能体。我们假设这些智能体为外部智能体，也就是说智能体没有能力对其它智能体的行为进行控制，它们有自己各自的目标并通过学习达到目标。对于外部智能体，我们对它们的目标、算法、协议、假设以及能力进行尽可能少的假设。

复杂环境是指具有巨大的或连续环境的参量，在这个环境中，相关的环境动力学依赖于连续或它们联合产生的特征。这种复杂性往往来自于环境中的其它智能体。例如，空间中的环境状态往往包括智能体自身的状态或位置。所以，环境的复杂性随着智能体数量的增加而增加，为了在这些领域中进行有效的行动，智能体对环境需要近似。

限制是阻碍智能体取得最优行动的限制条件，对于我们前面提到的复杂环境，限制是不可避免的。例如，近似限制了智能体对最优行动的选择。限制对所有的智能体都有影响，包括我们的智能体和其它智能体。在有限制条件下的智能体意味着都必须考虑智能体有没有能力采取最优行动。

由于以上提到的多智能体具有的优势和存在的问题，我们有必要对多智能体系统的理论和方法进行进一步的研究。

2 多智能体学习方法研究

2.1 多智能体学习的框架

框架是现实的模型。所以，框架是产生和评价新的思想的重要基础。框架产生“决策规则”，使得核心内容明确。框架提供了学习的基础，使得假设明了化，帮助对不同的解决方案进行分类，对巨大分类问题提供一般化的看法，对现实中的其它模型进行比较。由于以上原因，我们对多智能体学习的框架进行介绍。

首先我们从随机决策的框架开始。随机决策被认为是两种简单框架的结合：Markov 决策过程和双矩阵决策。如图 1 所示。Markov 决策过程在强化学习领域中得到了广泛的研究和探索，双矩阵决策是决策论的基础。Markov 决策过程是单智能体多状态的模型，而双矩阵决策是多智能体单一环境状态的模型。随机决策可以被看成是这两个过程的结合并包含这两个框架，定义了一个多智能体、多状态的框架。由于随机决策分享了这两个不同框架的概念，所以常常认为它们是不同的。

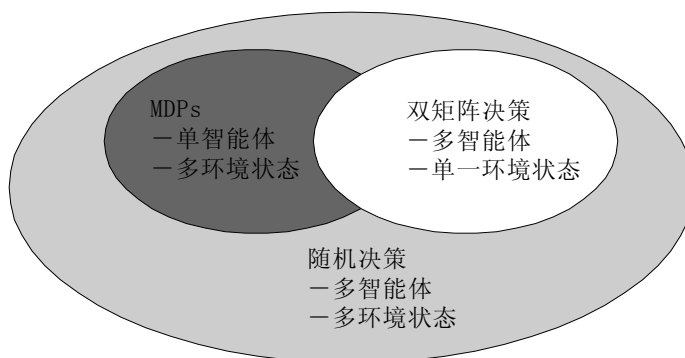


图 1 双矩阵决策、马尔可夫决策过程和随机决策关系图

所有的智能体都有三部分组成：感知、推理和行动。这三部分的具体操作过程如图 2 所示。智能体接受到环境的状态，从智能体可行行动域中选择行动。推理部分的任务是将接受到的环境状态映射到行动选择。智能体常常有几个目标，可能是从环境中期望得到的状态或一个最大化的信号，这是智能体集中学习的一部分。学习是智能体通过与环境的交互调整观察到行动的映射从而提高智能体达到目标的能力。

智能体的感知依赖于环境，而智能体采取的行动又会环境影响。本节中对环境进行了特别的定义：环境如何被智能体的行动影响，环境如何影响智能体的感知，是否有其它智能体存在。通过对环境不失一般性的假设，智能体对于要选择的期望行动进行有效的推理。

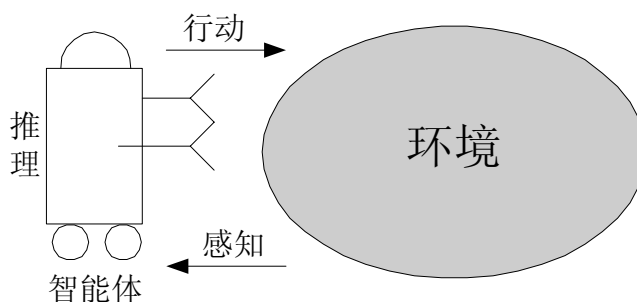


图 2 智能体模型

对于学习智能体，有两个另外的因素：第一，环境的详细状态是未知的。智能体只能通过与环境的不断交互获得有关环境的信息，也就是说，智能体通过选择行动并通过感知的输入观察行动的结果。第二，智能体接受额外的输入信号也

就是智能体得到的回报，这个回报取决于环境和智能体的行动。智能体的推理部分是一个学习过程，通过与环境的重复交互，随着时间以最大化它得到的回报。

学习框架考虑的另一个问题是有关智能体的观察。在本节中我们按常规的假设来定义智能体对环境的感知是完全的，也就是说智能体对环境的感知包括全部相关的环境状态。本节的工作是集中在存在其它智能体环境中智能体的学习的研究，图 3 描述了学习框图。不同于单一智能体在一个环境中的感知、推理和行动，环境中存在多个完全的智能体。这些智能体也在环境中感知、推理和行动。而且，它们也可能是学习智能体，通过与环境的交互适应它们的行动从而最大化它们的回报信号。

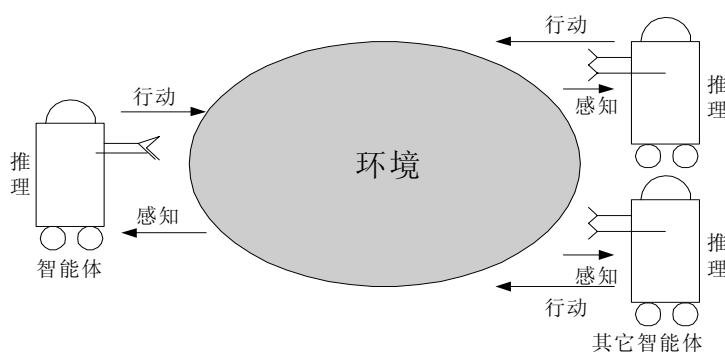


图 3 多智能体系统模型

三种形式的智能体和多智能体框架：Markov 决策过程、双矩阵决策和随机决策。Markov 决策过程是基本智能体框架。双矩阵决策考虑在单一环境中的多智能体框架，智能体的回报仅仅取决于智能体的行动。随机决策过程是全部多智能体框架，虽然我们的最终目的是集中在随机决策的包含模型，了解简单模型也是非常有用的^[39]。

2.2 双矩阵决策和马尔可夫决策过程

双矩阵决策^{[40][41]}是一种决策模型，在这个模型中，存在两个决策者 (Player)，它们同时选择动作策略并根据它们所选的动作对得到回报。

一个双矩阵决策由一个四元组 $\langle A^1, A^2, R^1, R^2 \rangle$ ，其中 A^i 为 Player i 的有限动作集合， R^i 为 Player i 的回报矩阵，当它的动作集为 (a^1, a^2) 而 Player 1 和 Player 2 选择的动作是 $a^1 \in A^1, a^2 \in A^2$ 时回报。Player i 的策略是用一个 A^i 的概率分布 π^i 来表示。

一个双矩阵策略的纳什平衡点是满足下列关系的策略对 (π_*^1, π_*^2)

$$(\pi_*^1)^T R^1 \pi_*^2 \geq (\pi^1)^T R^1 \pi_*^2 \quad \text{for any } \pi^1 \quad (3.1)$$

$$(\pi_*^1)^T R^2 \pi_*^2 \geq (\pi_*^1)^T R^2 \pi^2 \quad \text{for any } \pi^2 \quad (3.2)$$

已经证明, 对于任何一个有限的双矩阵决策存在至少一个纳什平衡点。如果一个平衡点 (π_*^1, π_*^2) 是确定的 (如 $\pi_*^i \in \{0,1\}, \forall a^i \in A, i=1,2$), 那么这个平衡点被称作纯策略。否则被称作混合策略的纳什平衡点。对于混合策略的纳什平衡点 (π_*^1, π_*^2) 满足下列的方程式。

$$(\pi_+^1)^T R^1 \pi_*^2 \geq (\pi_*^1)^T R^1 \pi_*^2 \quad (3.3)$$

$$(\pi_*^1)^T R^1 \pi_+^2 \geq (\pi_*^1)^T R^1 \pi_*^2 \quad (3.4)$$

其中 π_+^i 表示从动作集合中选取一个动作 $\{a^i \mid \pi_*^i(a^i) > 0, a^i \in A^i\}$ 的确定性策略。这一性质对于收敛到混合策略纳什平衡点的多智能体系统的设计是非常重要的。

$(\pi^1)^T R^i \pi^2$ 是 Player i 在 Player 1 和 Player 2 选取策略 π^1 和 π^2 时的期望回报, Player i 在纳什平衡点 (π_*^1, π_*^2) 称作 Player i 的平衡点值。

由于上一章已经对 Markov 决策过程进行了详细的介绍, 这里就不再赘述。如果使用值函数表示的话, Q 值函数被定义为

$$Q(s, a, \pi) = R(s, a) + \gamma \sum_{s' \in S} P(s' \mid s, a) v(s', \pi)$$

一个马尔可夫的最优策略是满足下式的策略 π_*

$$v(s, \pi_*) \geq v(s, \pi) \quad \forall \pi, s \in S$$

研究证明, 对于任何最优策略 π_* 下的值函数 $v(s, \pi_*)$ 是最优方程 (Bellman equations) 的唯一解^[42]。

$$v(s) = \max_{a \in A} \{ R(s, a) + \gamma \sum_{s' \in S} P(s' \mid s, a) v(s') \} \quad (3.5)$$

这个解 $v(s)$ 称作最优值函数, 任何有限马尔可夫决策过程有至少一个确定的最优策略, 所以对于单一智能体系统的强化学习无需处理概率性策略, 但对多智能体系统, 如果直接应用的话, 会存在问题^[43]。

2.3 随机决策

我们可以看到一个随机决策^{[44][45]}过程可以看作是一个马尔可夫过程在多智能体系统的扩展。过程如图 4 所示。

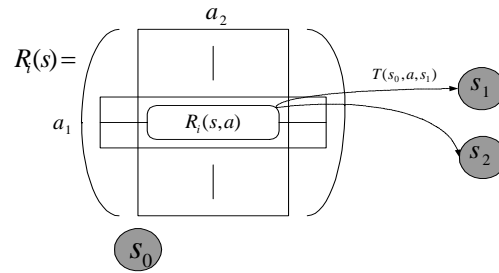


图 4 随机决策框架图

一个 2-Player、 γ 折扣的随机过程是一个六元组 $\langle S, A^1, A^2, P, R^1, R^2 \rangle$ 。 S 是一个有限状态集合， A^i 是一个智能体 i 有限的可能动作集合， $P: A^1 \times A^2 \times S \times S \rightarrow [0,1]$ 是转移函数， $P(s' | s, a^1, a^2)$ 是在状态 s 下， Player 1 和 Player 2 采用动作 a^1 和 a^2 时到达状态 s' 的概率。 $R^i: S \times A^1 \times A^2 \rightarrow \mathfrak{R}$ 是 Player i 的回报函数。 $R^i(s, a^1, a^2)$ 是在状态 s 下， Player 1 和 Player 2 采用动作 a^1 和 a^2 时 Player i 得到的回报。对于每一个智能体来说，其目标就是获得最大的总的折扣回报。

假设 Player I 的策略是 $\pi^i: S \times A^i \rightarrow [0,1]$ ，同时我们考虑静态策略，即在状态 s 下选择动作 a^i 的概率不随时间的变化而变化。对于给定的初始状态和策略 π^1 和 π^2 ，定义 Player I 的值函数为：

$$v(s, \pi^1, \pi^2) = \sum_{t=0}^{\infty} \gamma^t E(r^i_t | \pi^1, \pi^2, s_0 = s)$$

其中 $E(r^i_t | \pi^1, \pi^2, s_0 = s)$ 是 Player I 在时间 t 时的期望回报。所以 Q 函数可以写作：

$$Q_{\pi^1, \pi^2}^i(s, a^1, a^2) = R^i(s, a^1, a^2) + \gamma \sum_{s' \in S} P(s' | s, a^1, a^2) v^i(s', \pi^1, \pi^2) \quad (3.6)$$

一个随机决策的纳什平衡点是满足下列条件的策略对 (π^1, π^2) ：

$$v^1(s, \pi_*^1, \pi_*^2) \geq v^1(s, \pi^1, \pi_*^2) \quad \forall s \in S, \pi^1 \quad (3.7)$$

$$v^1(s, \pi_*^1, \pi_*^2) \geq v^1(s, \pi_*^1, \pi^2) \quad \forall s \in S, \pi^2 \quad (3.8)$$

已经证明，任何折扣有限随机决策至少有一个纳什平衡点，纳什平衡点是所有智能体的理性选择。所以多智能体系统强化学习希望收敛于纳什平衡点。

Filar 和 Vrieze 已经证明以下两种表述是等价的：

(π^1, π^2) 是一个平衡代价为 $(v^1(\pi^1, \pi^2), v^2(\pi^1, \pi^2))$ 的折扣随机决策平衡点，其中 $v^i(\pi^1, \pi^2) = (v^i(s, \pi^1, \pi^2))_{s \in S}, i = 1, 2$

对于每一状态 $s \in S$ ， $(\pi^1(s), \pi^2(s))$ 在静态双矩阵 $\langle A^1, A^2, Q^1_{\pi_1, \pi_2}(s), Q^2_{\pi_1, \pi_2}(s) \rangle$ 中组成平衡代价为 $(v^1(\pi^1, \pi^2), v^2(\pi^1, \pi^2))$ 的平衡点， $Q^i_{\pi_1, \pi_2}(s)$ 是人口 (a^1, a^2) 为 $Q^i_{\pi_1, \pi_2}(s, a^1, a^2)$ 的矩阵。

这条随机决策的平衡点性质对于多智能体强化学习系统的设计非常重要。

3 多智能体系统中的强化学习

由并发的强化学习者组成的多智能体系统在近年来吸引了最多研究者的注意力。多智能体的强化学习要比单智能体的强化学习困难得多。其难点在于，从一个智能体的角度来看，由于其它智能体的存在，其所处的环境不再是静态和确定的。当一个智能体与其它智能体存在利益冲突的时候，如果这个智能体仅仅是简单地对其它智能体进行最优响应地话，其适应性行为会引起其它智能体的行为改变，这样就意味着智能体所处的环境发生了变化，所以智能体又会去适应新的环境，使得智能体的行为策略陷入无尽的循环适应行为中。

正如马尔可夫决策过程（MDP）给单个智能体的强化学习提供了理论基础一样，随机决策为多智能体强化学习提供了理论基础。Littman^{[46][47][48][49]}基于这种理论框架提出了零和策略下的多智能体强化学习方法，Hu 和 Wellman^[50]将这种方法扩展到非零和决策。我们可以将 Hu 和 Wellman 的算法看作是单个智能体 Q-Learning 的扩展。一个进行 Q 学习的智能体其目标就是对环境状态的最佳响应。所以，在多智能体系统中，Q-Learning 可能会陷入无穷的适应性循环中。为了避免这种现象的发生，在 Hu 和 Wellman 的算法中，将智能体的学习目标定为对 Nash Equilibrium 的学习。这样的话，智能体在学习过程中就可以收敛到 Nash Equilibrium 上。但是从下面几方面来看，该算法又存在适应性不足的问题。智能

体把 Nash Equilibrium 作为学习目标而不考虑其它智能体的行为策略，因此如果其它智能体只采用固定策略而并非考虑 Nash Equilibrium 时，该算法的决策也许不是最佳决策。从另一方面来看，Hu 和 Wellman 的算法并没有考虑到 Nash Equilibrium 的选择问题，这就是说，即使所有的智能体都应用该方法来学习，在存在多个纳什平衡点的情况下，我们必须假设所有的智能体能够就纳什平衡点的选择达成一致意见，这种假设对于标准的自利型强化学习显然是不正确的。

在本文中，我们提出了一种新的多智能体强化学习算法。在算法中，我们仿照 Hu 和 Wellman 的算法，假设智能体能够识辨其它智能体的动作行为，并通过随机学习自动机对其它智能体的行动进行预测。在采用这种算法的情况下，如果其它智能体是适应性的智能体的话，我们的算法可以收敛，在其它智能体采用固定策略的话，我们的算法也可以得到对其它智能体行为策略的最佳响应。

3.1 智能体强化学习方法分类

我们将多智能体强化学习分成三种形式：合作型多智能体强化学习、竞争型多智能体强化学习和半竞争型多智能体强化学习。下面分别分析各自的特点和主要算法。

(1) 合作型多智能体强化学习

合作型多智能体强化学习中，由于在任意离散状态，马尔可夫对策的联合奖赏函数 R_i 对每个智能体来说是一致的、相等的，因此，每个智能体最大化自身期望折扣奖赏和的目标与整个多智能体系统的目标是一致的。事实上，并发独立强化学习和交互强化学习都属于合作型多智能体强化学习。在合作多智能体系统中，合作进化学习可以达到问题的最优解^[51]。

(2) 竞争型多智能体强化学习

在竞争型多智能体强化学习中，任意离散状态下马尔可夫对策的联合奖赏函数 R_i 对每个智能体来说是互为相反的。为叙述方便，我们以两个智能体为例，即系统中包含智能体 A 和智能体 B。图 3.5 给出两个智能体系统中某一状态下的对策模型。显然，该模型满足零和对策的定义：在任何策略下所有智能体的奖赏和为 0。

	Agent B	
	b_1	b_2

Agent A	a_1	(1 -1)	(4, -4)
	a_2	(2 -2)	(3, -3)

图 3.5 两个 Agent 零和对策模型

由于智能体 A 的奖赏取决于智能体 B 的动作，因此传统单智能体强化学习算法在竞争型多智能体强化学习中不适用。解决这一问题最简单的方法是采用极小极大 Q 算法：在每个状态 s ，对于智能体 A 其最优策略为智能体 B 选择最坏动作情况下，agent A 选择奖赏最大的动作。因此，定义竞争型多智能体强化学习的值函数为

$$V(s) = \max_{a \in A} \min_{b \in B} Q(s, a, b) \quad (3.9)$$

显然，如果将马尔可夫对策中每个状态都形式化为如图 3.5 的零合对策模型，那么极小极大 Q 算法可以发现最优策略。然而在竞争多智能体系统中，如果允许多个智能体同时进化，将导致系统非常复杂。Sandholm 等通过对追杀问题的实验表明，竞争进化学习 (Competitive Coevolution Learning) 不能够得到稳定解。但在竞争环境中，如果自身智能体不采用进化学习，而对手智能体采用进化学习，则任何稳定策略都会被击败。因此，在竞争多智能体系统中，需要对是否存在进化稳定策略或者智能体何时采用进化学习等问题作出明确的解释^[52]。

(3) 半竞争型多智能体强化学习

在许多实际多智能体系统中，往往单个智能体的所得奖赏并不是其它智能体所得奖赏和的负值，所以多智能体系统中离散状态 s 只能形式化为非零和对策。一个典型的示例是图 3.6 所示的囚犯两难问题。如果采用极小极大算法求解，其最优解为 (a_1, b_1) ，奖赏为 $(-9, -9)$ ；而显然囚犯两难问题的最优解为 (a_2, b_2) ，奖赏为 $(-1, -1)$ 。因此在非零和 Markov 对策模型中，用极大极小 Q 算法得不到最优解。

		Agent B	
		b_1	b_2
Agent A	a_1	(-9 -9) (-10 0)	(0, -10) (-1, -1)

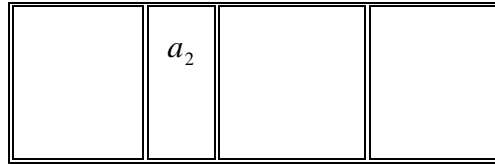


图 3.6 两个 Agent 非零和对策模型

本质上非零和对策模型更能反应多智能体系统中个体理性与集体理性冲突的本质^[53]。

3.2 Hu 和 Wellman 算法

Hu 和 Wellman 将单智能体 Q-Learning 算法推广到了多智能体系统^[54]。在他们的算法中,假设一个智能体可以感知其它智能体的行动和从环境中得到的回报,所有智能体的 Q 值表都被保存。在一些假设的前提下,当所有的智能体都按照这种算法进行决策的情况下,智能体的策略收敛到纳什平衡点。但是智能体往往不考虑其它智能体的策略就收敛到纳什平衡点。所以这种算法缺乏适应性。

在他们的算法中,当智能体在状态 s 选择动作 (a^1, a^2) 到达状态 s' 时,智能体得到回报, (r^1, r^2) , Q 值表通过下式进行更新:

$$Q^i(s, a^1, a^2) \leftarrow (1 - \alpha)Q^i(s, a^1, a^2) + \alpha\{r^i + \gamma V^i(s')\} \quad (3.10)$$

其中 $V^i(s')$ 是 Player I 在双矩阵决策 $\langle A^1, A^2, Q^1(s), Q^2(s) \rangle$ 的平衡点。策略 $\pi^1(s)$ 是 Player1 的策略,被设置为双矩阵策略 $\langle A^1, A^2, Q^1(s), Q^2(s) \rangle$ 的一个平衡点。因此,平衡点被评价状态到期望的策略应用。这也是这种算法缺乏对对手策略适应性的一个原因。

1 随机初始化 $Q(s \in S, a \in A)$, 设置学习率 α ;

2 重复

(a) 在状态 s 通过解矩阵决策 $u[Q(s, a)_{a \in A}]$ 选定动作 a , 同时采用一定的探索策略;

(b) 观察联合行动 a , 回报 r 和下一个状态 s' ,

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha\{r + \gamma V(s')\}$$

其中 $V(s) = \text{Value}([Q(s, a)_{a \in A}])$

对于 Littman 的最大最小 Q (Minimax-Q) 及 Hu 和 Wellman 的非零和纳什平衡点 (Nash-Q), 以上方法的不同在于值函数和 Q 值。Minimax-Q 利用线性规划的方法解决零和决策, 而 Nash-Q 利用二次规划解来解决非零和决策问题。

Nash-Q 是第一个考虑在非零和情况下随机决策的复杂问题的算法，但这个算法有几个限制性的假设条件。Nash-Q 算法中智能体环境状态随着其它智能体和自己的行动空间和状态空间的增加呈指数级增加，不但使学习系统设计难度提高而且使得系统的收敛速度急剧降低。

协作任务的强化学习算法中，常用的是独立强化学习算法。其特点是每个智能体不考虑其它智能体的行为，以自我为中心，进行各自的学习，在满足一定的条件下系统最终可以收敛，但需要的时间非常长。学习方法如下：

$$Q_t^k(s_t^k, a_t^k) = (1 - \alpha_t) Q_{t-1}^k(s_t^k, a_t^k) + \alpha_t [r_t^k + \beta \max_{a^k \in A^k} Q_{t-1}^k(s_{t-1}^k, a_t^k)]$$

其中：

s_t^k —— Agent k 在 t 时刻的状态；

a_t^k —— Agent k 在 t 时刻选择的动作；

r_t^k —— Agent k 在 t 时刻收到的强化信号；

α_t —— t 时刻的学习率；

$Q_t^k(s_t^k, a_t^k)$ —— Agent k 在 t 时刻的 Q 值。

4 基于 SLA 进行行动预测的多智能体强化学习算法

在经典的控制理论中，过程的控制是基于过程和系统完全知识的表达来完成的，假设数学模型、过程的输入和对时间的确定函数已知。后来控制理论的发展考虑到了系统的不确定性。随机控制过程假设有些不确定的特征是已知的。但是，所有这些都对不确定性或输入函数的假设，如果系统是变化的话，对系统的成功控制可能是不够的。所以需要在操作过程中观察过程并获得系统的进一步的知识，比如因为预先的假设不够，附加的信息必须通过在线学习获得。

虽然基于规则的系统在很多控制问题中都取得了满意的效果，但其缺点是即使在问题空间中一个很小的变化都需要调整。而且，基于规则的系统，特别是专家系统，不能处理非期望的情况。理想的学习系统的设计是在没有系统的完全信息的情况下保证行为的鲁棒性，与其它学习方法相比较，强化学习的最大优点是除了强化信号以外不需要其它的环境信息。

强化学习学习速度较其它方法慢是因为为了获得满意的操作效果，每一个动作都必须试验多次。随机自动机试一个问题的答案的时候不需要最优行动的任何信息（初始时，所有动作的概率都是相等的）。随机选择一个动作，观察环境的反应，动作的概率根据环境的反应进行更新，这种过程被不断重复。象前面所述

的提高操作水平的随机自动机被称为学习自动机。

学习自动机的学习环境样例可以用下面的描述：一个有限数量的动作可以在一个随机的环境中执行。当一个特定的动作执行后，环境的反应可能是正面的也可能是反面的，自动机设计的目标是确定在任何步骤过去的动作和反应如何指导动作的选择。最主要的一点是决策的确定对环境的自然状态需要很少的信息。环境也许是随时间变化的，决策的制定可能是分层决策结构但并不知道它在层次中的角色。更进一步说，环境的输出可能受其它智能体的影响而决策智能体却不知道。

自动机所处的环境对自动机行动的反应通过产生属于一系列许可的、可能与自动机有联系的反应。数学上，环境由一个三元组 $\{\alpha, c, \beta\}$ 来定义。 α 表示一个有限的动作/输出集， β 表示输入/反应集， c 是一个惩罚概率集，其中每一个 c_i 对应行动集合 α 中的一个行动 α_i 。

自动机的输出（动作） $\alpha(n)$ 属于动作集合 α ，在时间 $t = n$ 时作用于环境，环境的输入 $\beta(n)$ 是集合 β 的一个元素，可以表示为值 β_1 和 β_2 ，最简单的情形是 β_i 为 0 或 1，其中 1 表示失败或惩罚的反应。元素 c 定义为

$$\text{Prob}\{\beta(n) = 1 \mid \alpha(n) = \alpha_i\} = c_i \quad (i = 1, 2, \dots)$$

(3.11)

c_i 表示行动 α_i 得到环境惩罚的概率。当 c_i 是常数时，环境被称为是静态的。

人类个体通常只考虑自己周围的环境状态，同时根据对其他人可能采取什么行为的预测来决定自己的行为，别人所处状态仅用作预测的根据。多智能体强化学习系统中的智能体可以参照人类的决策过程来减少学习过程中需要考虑的因素，从而缩短学习过程。另外，由于所有智能体同时选择动作，这样在 t 时刻任一智能体都无法得知其它智能体将执行什么动作，所以智能体仍然无法根据来选择自己的动作。下面讨论的预测法可以用来实现协作型多智能体强化学习算法的动作选择策略，并能够加快学习的收敛速度^{[55][56][57]}。

4.1 基于 SLA 进行行动预测的多智能体强化学习算法

由于环境中多个智能体都同时执行动作而不会知道其它智能体执行什么动作，对第 i 个智能体 Agent i 可能执行动作的概率，其它智能体是通过基于 SLA 进行预测的，预测函数是 P ，它的更新规则如下

$$P_{t+1}^i(\mathbf{s}_t, a_k^i) = \begin{cases} P_t^i(\mathbf{s}_t, a_k^i) + \beta \sum_{a_i^i \in A - \{a_k^i\}} P_t^i(\mathbf{s}_t, a_i^i) & , \text{if } a_k^i = a_i^i \\ (1 - \beta) P_t^i(\mathbf{s}_t, a_k^i) & , \text{otherwise} \end{cases} \quad (3.11)$$

其中：

$P_{t+1}^i(\mathbf{s}_t, a_k^i)$ —— Agent i 在时刻 t+1 可能执行动作的概率预测函数

$$\sum_{a_k^i} P_t^i(\mathbf{s}_t, a_k^i) = 1$$

a_k^i —— Agent i 动作集中第 k 个动作；

β —— 预测模型的学习率；

式(3.13)就是一个随机学习自动机，根据式(3.13)更新的概率预测函数 P ，在采用标准 Q-learning 的强化学习的情况下，其它 Agent j 的 Q 函数更新规则为：

$$Q_t^j(s_t^j, \mathbf{a}_t) = (1 - \alpha) Q_{t-1}^j(s_t^j, \mathbf{a}_t) + \beta \pi^1(\mathbf{s}_{t-1}) \cdots \pi^n(\mathbf{s}_{t-1}) Q_{t-1}^j(s_{t-1}^j)$$

其中：

$$\begin{aligned} \pi^1(\mathbf{s}_{t-1}) \cdots \pi^n(\mathbf{s}_{t-1}) Q_{t-1}^j(s_{t-1}^j) &= \sum_{a^1 \in A} \sum_{a^2 \in A} \cdots \\ &\sum_{a^n \in A} P_t^1(\mathbf{s}_{t-1}, a^1) \cdots P_t^n(\mathbf{s}_{t-1}, a^n) Q_{t-1}^j(s_{t-1}^j, a^1, a^2, \cdots, a^n) \end{aligned}$$

(3.12)

多智能体强化学习算法的动作选择过程如下：

首先得到

$$Q(s^1, a^1, a^2, \cdots, a^n, s^{n-1}) = \sum_{a^n \in A} P^n(\mathbf{s}, a^n) Q(s^1, a^1, a^2, \cdots, a^n)$$

依次下去，最后得到

$$Q(s^1, a^1) = \sum_{a^2 \in A} \sum_{a^3 \in A} \cdots \sum_{a^n \in A} P^2(\mathbf{s}, a^2) P^3(\mathbf{s}, a^3) \cdots P^n(\mathbf{s}, a^n) Q(s^1, a^1, a^2, \cdots, a^n)$$

然后用 Boltzmann 机根据 $Q(s_1, a_1)$ 选择 a_1 ，如下式：

$$prob(a^1) = \frac{\exp(\gamma Q(s_1, a_k^1) / T)}{\sum_{a_t^1 \in A^1} \exp(\gamma Q(s_1, a_t^1) / T)}$$

(3.13)

其它智能体的动作选择过程与此类似，Hu 已经证明多智能体强化学习算法的收敛性与其动作选择机制无关，所以用以上方法进行动作选择不会影响多智能体强化学习算法的收敛性。

4.2 多机器人推箱子问题

为了研究强化学习在多智能体系统中的应用，我们采用了一个多机器人协作推箱子的问题，试验环境如图 8。

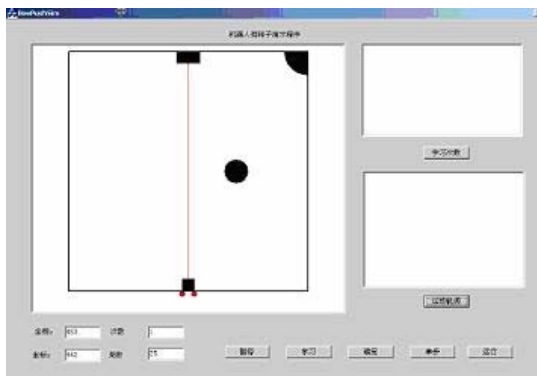


图 7 试验环境

在这个环境中， $a_i (i=1,2,\dots,n)$ 为 n 个独立的机器人，它们的任务是将箱子 B 从一个起始点 S 寻找最佳路径推到一个目标位置 G 。这些机器人相互之间并不清楚其它机器人的能力或当前将要执行的动作，所以它们必须独立地或通过对其它机器人可能采取行动的预测选择自身的动作来完成协调任务。系统中，机器人能够感知其它智能体的动作及自身所处的环境状态。

机器人 i 在角度 θ_i 给箱子一个力 $\vec{F}_i (0 \leq \vec{F}_i \leq F_{\max})$ ，其中 $0 \leq \theta_i \leq \pi$ 。箱子在 x 方向移动 $\vec{F}_i \cos(\theta_i)$ 的距离，在 y 方向移动 $\vec{F}_i \sin(\theta_i)$ 的距离。多机器人作用于箱子的合力是向量和 $\vec{F} = \vec{F}_1 + \vec{F}_2 + \dots + \vec{F}_n$ 。对箱子新位置的计算我们假设单位的力将使箱子移动单位的距离。其作用力图 8 所示。

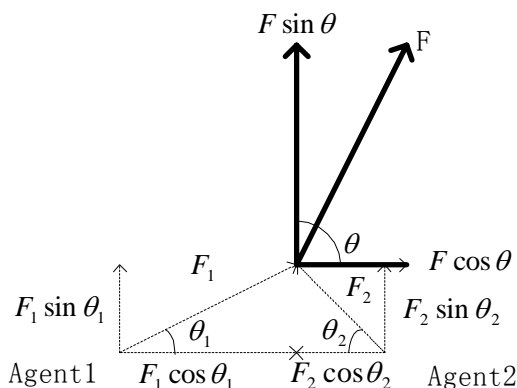


图 8 推箱子问题力学模型

箱子的位置将作为参数反馈给每个机器人。假设 (x, y) 是箱子的当前位置, (x_g, y_g) 表示目标坐标, 那么 $\Delta d = \sqrt{(x_g - x)^2 + (y_g - y)^2}$ 就是箱子的当前位置与给定目标位置的偏差。环境的回报值包括两个部分: $K * \alpha^{-\Delta d}$ 作为与目标的偏离程度报酬, r_p 为箱子与障碍物碰撞得到的惩罚, 将这两部分反馈给每个机器人的作为环境回报值。

学习效率我们采用递减的方法, 学习效率 β 取初值为 1.0, 其递减方法为 $\beta = 0.99\beta$, 为了加速系统的收敛, 探索的措施采用 $\epsilon - greedy$ 方式, ϵ 取初值 0.5, 随着学习效果的取得, 我们逐渐降低探索率, 因此对探索率也采用递减的方法 $\epsilon = 0.99\epsilon$ 。通过试验验证, 证明这两种措施是有效的, 加快了系统的收敛速度, 缩短了系统的学习时间。

试验系统是在一个 400×400 的正方形平台上进行, 试验中一局的制定是指机器人将一个箱子从起点 S 推到目标点 G、箱子被推出场外或与障碍物碰撞。我们设置一局在试凑一定的次数后进行强制停止, 以免学习系统陷入极点。虽然在试验中我们仅仅用了两个机器人, 但此方法同样适用于更多的机器人协作系统。

4.3 试验及结果比较

我们对以下两种算法进行了试验:

- (1) 机器人独立行动强化学习算法;
- (2) 采用本文提出的基于预测的强化学习算法。

对图 3.7 的试验环境, 为验证算法的有效性, 我们设计了三种试验环境: 直线无障碍环境、斜线无障碍环境和斜线有障碍环境, 并对斜线有障碍环境的试验结果进行了比较。

图 9 试验得到的箱子最佳路径。图 10 为本算法与独立强化学习算法收敛速度比较, 图 11 算法中行动预测准确性。从得到的结果来看, 本章提出的算法能够较快地收敛于最佳路径, 随着训练过程的进行, 一个机器人对另一个机器人动作预测的准确度逐渐提高。试验结果验证了我们提出的强化学习算法在多智能体协作应用中的有效性。

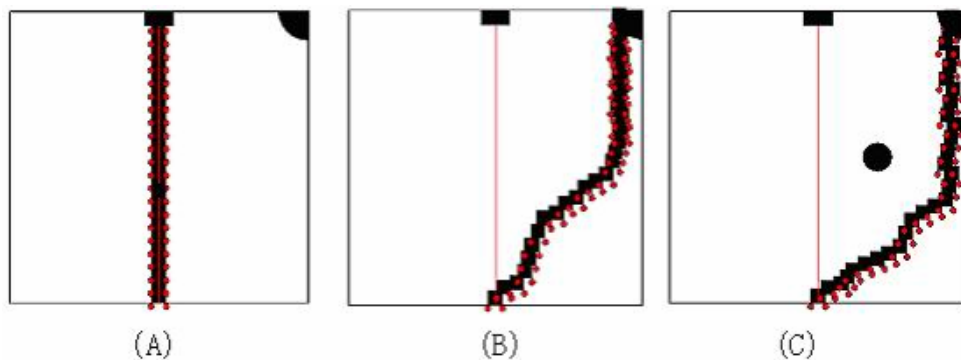


图 9 试验得到的箱子最佳路径((A)直线无障碍环境(B)斜线无障碍环境(C)斜线有障碍环境)

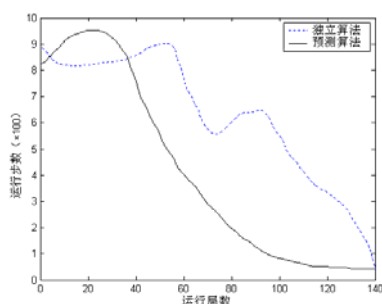


图 10 本算法与独立强化学习算法收敛速度比较

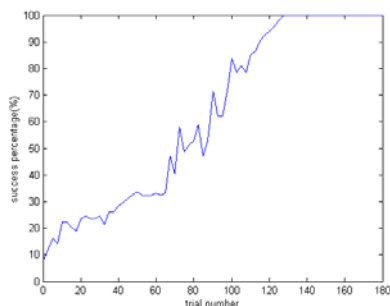


图 11 算法中行动预测准确性

5 小结

本章对多智能体系统的理论框架及理论基础进行了详细的研究,结合已有的多智能体强化学习方法和随机学习自动机理论,提出了一种基于 SLA 进行行动预测的多智能体强化学习方法。

在多智能体系统中,评价一个智能体行为的好坏常常依赖于其它智能体的行为,此时必须考虑其它智能体的行为。对于已有的多智能体强化学习算法,大部分是集中于基于纳什平衡点原理的群体强化学习算法或独立强化学习算法。前者

采用组合动作(所有 Agent 的动作组成的动作向量)来保证整个团队协调地完成预定任务,而且要求系统具有较强的通讯能力,但由于采用组合动作使智能体环境状态数量变得巨大而收敛得极慢。独立强化学习没有考虑其它智能体的动作行为,系统中的 Agent 都以自我为中心,所以不易达成协作也使学习时间加长。本文提出基于 SLA 来预测各机器人执行动作的概率而不采用行动组合,降低了学习空间的维数,同时考虑了其它智能体的行为概率,加快了系统的收敛速度。并通过多机器人协作推箱子问题对算法进行了试验验证。结果表明,算法与现有的多智能体强化学习方法相比,其收敛速度有较大的提高,证明了算法的有效性。

参考文献

-
- [39] Bellman,R. Dynamic programming.Princeton University Press,1957.
- [40] T. Basar and G. J. Olsder, Dynamic Noncooperative Game Theory. London: Academic Press, 1982.
- [41] B. Banerjee and J. Peng, Adaptive policy gradient in multiagent learning, in Proceedings of the second international joint conference on Autonomous agents and multiagent systems.ACM Press, pp. 686–692, 2003.
- [42] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, Massachusetts: MIT Press, 1998,1990.
- [43] M. L. Littman, Value-function reinforcement learning in markov games, Journal of Cognitive Systems Research, vol. 2,pp. 55–66, 2001.
- [44] D. Fudenberg and J. Tirole, Game Theory. London: MIT Press,1991.
- [45] Gullapalli,V.A stochastic reinforcement learning algorithm for learning real-valued functions.Neural Networks 3:671-692,1992.
- [46] M. L. Littman and P. Stone, Leading best-response strategies in repeated games, in The 17th Annual International Joint Conference on Artificial Intelligence Workshop on Economic Agents, Models, and Mechanism, 2001.
- [47] M. L. Littman, Friend-or-foe Q-learning in general-sum games, in Proceedings of The 18th International Conference on Machine Learning, Morgan Kaufman, pp. 322–328, 2001.
- [48] M. L. Littman, Value-function reinforcement learning in markov games, Journal

of Cognitive Systems Research, vol. 2, pp. 55–66, 2001.

- [49] M. L. Littman, Markov games as a framework for multiagent learning, in Proceedings of the Eleventh International Conference on Machine Learning, San Francisco, California, pp. 157–163, 1994.
- [50] J. Hu and M. P. Wellman, Multiagent reinforcement learning in stochastic games, [Online]. Available: citeseer.ist.psu.edu/hu99multiagent.html, 1999.
- [51] Narendra P, Sandip S, Maria Gordin, Shared memory based cooperative coevolution. In Proceedings of the 1988 IEEE International Conference on Evolutionary Computation. Alaska, USA: IEEE Press, 570-574, 1998.
- [52] Tuomas W Sandholm, Robert H Crites. On multiagent Q-Learning in a semi-competitive domain. In: Weiss G, Sen S, Adaption and learning in Multiagent System. Lecture Notes in Artificial Intelligence, 1024. NY, Springer, 191-205, 1996.
- [53] Takuya Ohko, Kazuo Hiraki, Yuichiro Anzal. Learning to reduce communication cost on task negotiation among multiple autonomous mobile robots. In: Weiss G, Sen S, Adaption and learning in Multiagent system, Lecture Notes in Artificial Intelligence, 1042. NY: Springer, 177-190, 1996.
- [54] Hu and M. P. Wellman, Nash Q-learning for general-sum stochastic games, Journal of Machine Learning Research, vol. 4, pp. 1039–1069, 2003.
- [55] Littman M, Szepesvari C. A generalized reinforcement learning model: Convergence and applications [A]. Proceedings of the 13th International Conference on Machine Learning. Bari, Italy, July 3-6: 310-318, 1996.
- [56] Junling Hu, Michael P Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. Proceedings of the 15th International Conference of Machine Learning [C]. July 24-27, Madison Wisconsin: 115-122, 1998.
- [57] J.-H. Kim and P. Vadakkepat, Multi-agent systems: a survey from the robot-soccer perspective, International Journal of Intelligent Automation and Soft Computing, vol. 6, no. 1, pp. 3–17, 2000.