多 Agent 系统中强化学习的研究现状和发展趋势*)

赵志宏 高 阳 骆 斌 陈世福
(南京大学计算机软件新技术国家重点实验室 南京210093)
(南京大学计算机科学与技术系 南京210093)

摘 要 本文对有关强化学习及其在多 Agent 系统中的应用等方面的研究现状、关键技术、问题和发展趋势进行了综述和讨论,试图给出强化学习目前研究的重点和发展方向。主要内容包括:(1)强化学习的框架结构;(2)几个有代表性的强化学习方法;(3)多 Agent 系统中强化学习的应用和问题。最后讨论了多 Agent 系统中应用强化学习所面临的挑战。

关键词 多 Agent 系统,强化学习,算法

Reinforcement Learning Technology in Multi-Agent System

ZHAO Zhi-Hong GAO Yang LUO Bin CHEN Shi-Fu
(State Key Laboratory for Novel Software, Nanjing University, Nanjing 210093)
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract This paper introduces the key techniques, the current state of the research and the development tendency of the reinforcement learning and its applications in the multi-agent system. The content includes: (1) the structure of the reinforcement learning; (2) several representative reinforcement learning algorithms; (3) the applications and problems adopting reinforcement learning algorithms in the multi-agent system. Finally, the challenge that will be encountered when adopting reinforcement learning algorithms in the multi-agent system is discussed.

Keywords Multi-agent system, Reinforcement learning, Algorithm

1 引言

Agent 的概念可以追溯到70年代对 DAI 的研究。1977年,Car Hewitt 在 Actor 模型中定义:"Actor 是一个封装了地址和行为的计算 Agent,Actor 通过消息传递进行通信,并且并发地执行动作"[1]。Actor 模型被认为是 Agent 概念的原型。目前,对于何谓 Agent 这个问题有多种看法,一个较为普遍的观点是:Agent 是一个具有自主性、社会能力和反应特征的计算机软、硬件系统,即它的主要特征是具有自治性、社会能力、反应性和主动性[2]。此外,常被讨论的其他特点还有移动性、诚实性、善意性等等。多 Agent 系统(MAS)即由多个 Agent 构成的系统,可泛指所有由多个自治或半自治模块组成的系统。

多 Agent 系统的学习并不是单 Agent 学习的简单增强,事实上,多 Agent 系统的学习过程是相当复杂的,直接依赖于多个 Agent 的存在和交互。因此,多 Agent 系统的学习又被称为交互学习[3]。在协商策略的研究中,对于 Agent 如何得到合适的策略有两种解决方案,一种方案是在 Agent 初始化时配置足够的策略,这意味着需要先验描述所有可能的情况及其解决办法,对于大多数的实际应用来说这是不可能做到的;另外一种方案是 Agent 自身具有学习能力,能够从协商过程中获取经验。Dworman 等人[4]指出,在当前还不足以成功地将人类的协商技巧程序化的情况下,Agent 在协商过程中通过学习不断更新策略是一个有效可行的办法。

强化学习是从环境状态到行为映射的学习,以使 Agent 的累计奖赏值最大。采用强化学习机制的 Agent 通过尝试来选择具有最大累计奖赏值的行为策略[5],这意味着系统设计者只需给出最终需要实现的目标,而不需指出如何去达到目标[6]。正因为具有以上的特性,使得强化学习成为多 Agent 系统协商中更新行为策略的一类重要算法。本文第2节描述了强化学习的基本框架,第3节介绍了目前较为常见的强化学习算法,第4节介绍了将强化学习算法应用到多 Agent 系统的各类对策机制。

2 强化学习的基本框架

强化学习技术是从控制论、统计学、心理学等相关学科发展而来的,有着相当长的历史,但直到20世纪80年代末、90年代初强化学习技术才在人工智能、机器学习中得到广泛研究,并被认为是设计智能 Agent 的核心技术之一[7]。

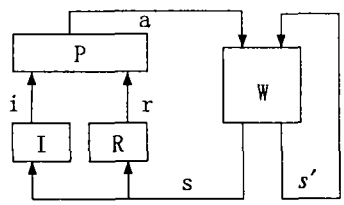

图1 强化学习基本框架

*)本文得到国家自然科学基金(项目编号:60003010,69905001)的资助。赵志宏 博士生,主要研究方向为多 Agent 系统,强化学习,数据库技术。高 阳 博士,讲师,主要研究方向为多 Agent 系统,强化学习。骆 斌 博士,副教授,主要研究方向为多 Agent 系统,数据库技术。陈世福 博士生导师,教授,主要研究方向为人工智能。

标准的强化学习框架如图1所示。在 Agent 与环境每一次交互过程中，接受环境状态 s 的输入，并映射为 Agent 的感知 i，Agent 选择行为动作 a 作为对应环境状态的输出。行为动作 a 将导致环境状态 s 变迁到 s'，同时 Agent 接受环境的奖惩信号 r。Agent 的目标是在每次选择行为时，使选择的行为能够获得环境最大的奖赏 v。如果 Agent 的某个行为策略 π 能够获得环境的奖赏，那么 Agent 以后策略 π 的趋势便会加强，这是一个正反馈的过程。标准的 Agent 强化学习框架由三个模块组成：输入模块 I、强化模块 R 和策略模块 P。其中输入模块 I 将环境状态映射为 Agent 的感知 i，强化模块 R 根据环境状态 s 到 s' 的迁移来赋予 Agent 奖赏值 r；策略模块 P 更新 Agent 的内部知识，同时使 Agent 根据某种策略选择一个动作作用于环境。

若定义 S 为环境所有可能状态的集合，X 为 Agent 所有感知的集合，A 为 Agent 的行为集合。则 Agent 可以用三元组 (I,R,P) 描述。其中：

$I:S\to X$　$R:S\to\mathscr{R}$　$P:X\times R\to A$

环境状态转移函数 W 可定义为 $W:S\times A\to S$

强化学习需要定义一个目标函数来评估从长远看动作是否是最优的。通常以状态的值函数(Value function)或状态-动作对的值函数体现此目标函数，其目标函数的形式有以下三种：

$$V=\sum_{t=0}^{\infty}\gamma^t r_{t+1};\ V=\sum_{t=0}^{h} r_t;\ V=\lim_{h\to\infty}\left(\frac{1}{h}\sum_{t=0}^{h} r_t\right)$$

(其中$0\leqslant\gamma\leqslant 1$称为折扣因子)

分别称为无限折扣模型，有限折扣模型和平均奖赏模型。无限折扣模型考虑未来的奖赏，有限折扣模型仅考虑未来 h 步的奖赏，平均奖赏模型则着重于考虑其长期平均奖赏。

3 常见的强化学习算法

一般在顺序性任务中，常采用 Markov 决定过程(MDP)进行建模，通过著名的 Bellman 公式进行迭代求解[8]。对于有限 MDP，Bellman 优化方程存在与策略无关的唯一解。采用 Markov 决定过程建模的强化学习算法可分为两类。一类强化学习算法先进行模型的学习，再根据模型知识推导优化策略，这类算法被称为基于模型法(Model-based)；另一类强化学习算法直接计算优化策略，这类算法被称为模型无关法(Model-free)。下面我们将着重介绍几种常见的强化学习算法：

3.1 TD 算法

一步 TD 算法，即 TD(0)算法，是一种自适应的策略迭代算法，又名自适应启发批评算法(Adaptive Heuristic Critic，AHC)。该算法是由 Sutton 于1988年提出的[9]，并由 Sutton 首先证明了其在特定条件下的收敛性[10]。Jaakkola，Jordan 及 Singh[7]对该算法作出了扩充。所谓一步 TD 算法，是指 Agent 获得的瞬时奖赏值仅回退了一步，也就是说只是修改了相邻状态的估计值。TD(0)算法如下：

$$V(s)=V(s)+\alpha(r+\gamma V(s')-V(s))$$

其中 α 为步长，$V(s)$ 指在环境状态 s 下获得的奖赏和，$V(s')$ 指环境状态转移到 s' 下时获得的奖赏折扣和。

TD(0)算法可扩充到 TD(λ)算法，即 Agent 获得的瞬时奖赏值可回退任意步。TD(λ)算法的收敛速度有很大程度上的提高，算法可形式化如下：

$$V(u)=V(u)+\alpha(r+\gamma V(s')-V(s))e(u)$$

其中 e(u)定义为状态 u 的选举度。

$$e(s)=\sum_{k=1}^{t}(\lambda\gamma)^{t-k}\delta_{s,s_k}$$，当 $s=s_k$，δ_{s,s_k}为1，否则为0

$$e(s)=\begin{cases}\gamma\lambda e(s)+1 & \text{若 } s \text{ 为当前状态}\\ \gamma\lambda e(s) & \text{其他}\end{cases}$$

3.2 Q-学习

Q-学习是一种与模型无关的强化学习算法。Q-学习迭代时采用环境-动作奖赏和 $Q^*(s,a)$作为衡量标准，而非 TD 算法中的环境奖赏和 $V(s)$，这样在每一次迭代中都需要考察 Agent 的每一个行为，可保证其收敛。Q-学习由 Watkins 提出，Watkins 和 Dayan 于1992年基本证明了其收敛性[11]，其后 Jaakkola，Jordan，Singh 和 Tsitsiklis 又分别于1994年作出了更加详细的泛化证明[12]。Q-学习也可根据 TD(λ)算法的方式扩充到 Q(λ)算法。Q-学习算法的基本形式如下：

$$Q^*(s,a)=R(s,a)+\gamma\sum_{s'\in S'}T(s,a,s')\max_{a'}Q^*(s',a')$$

其中 $Q^*(s,a)$表示 Agent 在状态 s 下采用动作 a 所获得的奖赏折扣和。由此可知，最优策略为在 s 状态下选用 Q 值最大的行为。Q-学习虽然需要提高收敛速度，但由于在一定条件下 Q-学习只需采用贪心法(greedy method)即可保证收敛，因此 Q-学习是最有效的模型无关强化学习算法之一。

3.3 Sarsa

Sarsa 算法是 Rummery 和 Niranjan 于1994年提出的[13]，它是一种模型相关算法，最初被称为改进的 Q-学习算法。一步 Sarsa 算法被 S. Singh 证明是收敛的。Sarsa 算法主要考虑的是行为值函数 $Q(s,a)$而非状态值函数 $V(s)$。一步 Sarsa 算法可用下式表示：

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

其中 t 和 t+1为时刻。上式显示了一步 Sarsa 算法的迭代过程。可以明显看出 Sarsa 与 Q 的差别就在于 Q 学习采用的是最大值进行迭代，而 Sarsa 则采用的是实际的 Q 值进行迭代。

3.4 Monte-Carlo 方法

Q(0)算法中是基于迁移前观察到的最大值来推定 Q 值的，然而在 Monte-Carlo 方法中，算法是以各时刻 t 所得到的实际报酬为基础来推定 Q 值，而不使用状态迁移前的 Q 值[14]。这一点不但与 Q(0)不同，与 Sarsa(0)也不同。采用这种机制能够使采用了错误的状态转移概率带来的不良影响降到最低。在非 Markov 状态下，这种性质是相当重要的。

3.5 Dyna 算法

Dyna 算法是一个复合体，它综合了动态规划和 TD 算法，Agent 通过三步学习优化策略。首先 Agent 使用学习经验来简历模型，其次是用经验来调节策略，最后使用模型来调节策略。Dyna 算法由 Sutton 于1991年提出[15]，主要的目的是为了充分利用每次学习经验中获取的知识，从而解决 TD 算法和 Q-学习迭代速度较慢的问题。

4 多 Agent 系统中强化学习的应用和问题

从机器学习角度来看，有两种方式可将强化学习技术应用于多 Agent 系统，其一是将多 Agent 系统作为一个可计算的学习 Agent，其二是每个 Agent 有自己的强化学习机制，通过与其他 Agent 适当交互加快学习过程。1993年，Tan[16]在论文中提出将强化学习机制应用于猎人合作游戏，多个猎人共享感知、经验和策略，从而更好地完成任务；1995年，Pan[17]提出了一个分布式强化学习系统框架，该环境由一个用于处理不完全感知信息的隐含任务模型 HMM，一个相关任务的组合状态模型 CSM 和一个进行特性修改的 Q-学习系统 QLS

组成。除此之外，研究人员还将多 Agent 系统中强化学习机制应用到了其他各个领域，例如邮件路由选择[18]、口语对话系统[19]以及机器人足球赛[20]等等。在这些应用中，研究人员广泛采用了各种对策论机制来对多 Agent 系统的策略进行控制，协调和评估。最先提出的，也是应用最为广泛的机制是 Markov 对策（Markov Game），又称随机对策（Stochastic Game）。值得一提的是，尽管目前针对动态多 Agent 环境下强化学习的绝大多数讨论均基于 Markov 对策，但也有一部分讨论加入了元对策理论（Meta Game）。元对策理论由 Thomas 于1984年提出[21]，其中心思想是在已知非零和对策模型的情况下，通过分析 Agent 自身的愿望和预测对手的策略来修正自己的策略，从而得到全局最优解。元对策避免了求解复杂的 Nash 平衡解[22]。鉴于在元对策和其他对策理论基础上的讨论并不多，本文不再详述。

4.1 Markov 对策理论及其解决方案

定义1（Markov 对策） 离散的状态集 S，Agent 动作集 A^i 的集合 A，联合奖赏函数 $r_i: S\times A^1\times A^2\times\cdots\times A^n\to R$ 和状态转移函数 $T: S\times A^1\times A^2\times\cdots A^n\to PD(S)$。Agent 的目标是在每个离散状态 s 寻求最优策略 π 以使期望的折扣奖赏和 $v^i(s,\pi^1\times\pi^2\times\cdots\times\pi^n)$ 最大，其中 $v^i(s,\pi^1\times\pi^2\times\cdots\pi^n)$ 定义如下：

$$v^i(s,\pi^1,\pi^2,\cdots,\pi^n)=\sum_{t=0}^{\infty}\gamma^t E(r_t^i\mid\pi^1,\pi^2,\cdots,\pi^n,s_0=s)$$

上式中每个 Agent 获得的瞬时奖赏都依赖于其他 Agent 的动作。t 时刻所有 Agent 获得的奖赏满足 $\sum_i r^i(s,a^1,a^2,\cdots,a^n)=0$ 的对策是零-和 Markov 对策，否则即是非零-和 Markov 对策。

在 Markov 对策理论的框架下，研究人员提出了各种解决方案。其中有代表性的方案包括：

·*直接对单 agent 系统进行扩展* 这也是最初提出的方案之一。将整个系统中的其他 agent 均看作是环境的一部分，直接将单 Agent 的强化学习算法扩展到多 Agent 系统[23]，这类算法常常导致不收敛的结果。

·Minimax Q-learning 针对动态多 Agent 环境，Littman 选取了1对1的足球 game 作为零-和 Markov 对策的例子，提出了 Minimax Q-learning[18]。算法的核心思想是：对于 Agent a 和 Agent o，最优策略为 $V(s)=\max_{a\in A}\min_{o\in O}Q(s,a,o)$。Minimax Q-learning 仅能解决零-和 Markov 对策问题，适用范围较窄。同时，Bowling 和 Veloso 也指出，该算法的一个问题在于尽管能够取得收敛的解，但是取得的解往往并非有价值，在石头-剪刀-布游戏中尤为明显[24]。

·*基于 Nash 平衡解的算法* 由于 Minimax Q-learning 算法仅能解决零-和 Markov 对策问题，因此在求解非零-和 Markov 对策时，常采用 Nash 谈判公理求平衡解。非零-和 Markov 对策的 Nash 平衡解定义如下：

定义2 2-player Markov 对策 $\Gamma=\langle S,A^1,A^2,r^1,r^2,T\rangle$ 的 Nash 平衡解[25]是满足以下条件的策略对 (π_*^1,π_*^2)：

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

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

其中策略 $\pi=(\pi_0,\cdots,\pi_t,\cdots)$ 是定义在整个对策空间上的，π_t 是时刻 t 的决策规则。满足条件 $\pi_t=\bar{\pi}$ 的策略称为静态策略；满足条件 $\pi_t=f(h_t)$ 的策略称为行为策略，h_t 是到时刻 t 为止的历史：$h_t=(s_0,a_0^1,a_0^2,s_1,a_1^1,a_1^2,\cdots,a_t^1,a_t^2,s_t)$。

Filar 和 Vrieze 证明了在非零-和折扣 Markov 对策中至少存在一组静态策略满足 Nash 平衡解的定义[26]。直接求解 Nash 平衡解的困难在于其计算复杂性过大，因此难以得到广泛运用。

1998年，Hu 等人用非零-和对策作为描述多 Agent 强化学习的理论框架，并设计了可收敛到 Nash 平衡解的多 Agent 强化学习算法[27]，将 Minimax Q-learning 扩展到处理非零-和 Markov 对策，并证明了算法必然收敛到 Nash 平衡解。不过，该算法不一定能收敛到最佳解、或和目标相关的解，在石头-剪刀-布游戏中 Minimax Q-learning 表现出的问题也同样存在于 Hu 的算法中[24]。同时，算法采用 Q 值表，要求系统保留较大的存储空间，这也限制了它的应用。

2001年，Bowling 和 Veloso 在文[24]中提出了评价多 Agent 强化学习算法的两个准则，即收敛性准则和合理性准则。文中定义了以"得到和目标达成相连的解"作为多 Agent 强化学习算法的衡量条件之一，提出了保证合理性同时也保证收敛性的算法 WoLF Policy Hill Climbing（PHC），并在文[28,29]中证明了该算法的收敛性。

4.2 多 Agent 系统强化学习研究的方向

多 Agent 系统强化学习中可研究的课题很多，例如不完全感知问题、同时学习问题、求解的计算复杂性问题、报酬分配问题以及与其他技术的结合问题等等。限于篇幅，本文将重点介绍目前研究得较多的不完全感知问题、同时学习问题和报酬分配问题，其他问题只进行简要的介绍。

（1）*不完全感知问题* 一般在有着大规模的状态空间的多 Agent 系统中，不完全感知问题是无法避免的。不完全感知问题研究的是当 Agent 处于无法直接通过观察环境或其他手段了解到全部的信息时，系统将产生一系列问题。由于 Agent 所获得的信息不完整，因而将导致错误的状态迁移，从而影响到状态转移概率。单 Agent 的不完全感知问题一般采用 POMDP 模型进行讨论。在运筹学领域中，POMDP 早已得到广泛的研究与应用[30,31]，但将其应用到 Agent 及多 Agent 系统中则是近年来才开始的。1994年，Singh，Jaakkola 和 Jordan 提出通过观察-动作映射的形式来描述策略[32]，在他们的论文中采用了概率分布而非确定的动作来作为选择动作的依据。POMDP 模型是在他们工作的基础上由 Kaelbing，Littman 和 Cassandra 提出的[33]，其定义如下：

定义3 一个 POMDP 模型可用六元组 $\langle S,A,T,R,\Omega,O\rangle$ 表示，其中 S,A,T,R 定义了一个 MDP，Ω 表示有限的观察集合，$O: S\times A\to\Pi(\Omega)$ 为观察函数，表示在给定动作和状态的情况下可能得到的观察的概率分布。

POMDP 模型可在信念状态空间中转化为 MDP 进行求解。Sondik 和 Smallwood 在20世纪70年代即已阐明：POMDP 模型的值函数是分段线性凸函数，可以通过值迭代的方式进行求解[34,35]。但是，在求解 POMDP 模型时，随着迭代次数的增加，其值函数有很高的计算复杂度，对于无限阶段折扣模型甚至有可能是不可计算的[36]。为解决这个问题，研究人员进行了大量的工作，有代表性的是采用启发式近似方法来进行计算[37]，最简单的是 MDP 近似算法，还有 QMDP 算法[38]和基于线性函数更新的格点近似算法[39,40]等等。对 POMDP 模型的研究目前正受到越来越多的重视[41,42]。在多 Agent 系统中采用 POMDP 模型，协调的难度和计算复杂性将更大。

从另一个角度来看，在多 Agent 系统中，单纯地将其它 Agent 当作环境的一部分，在信息不共享的情况下，可使用 Monte-Carlo 方法、Sarsa（$\lambda>1$），或报酬分配法来解决不完全感知问题。Arai 指出：将合理的报酬分配方法应用于各

Agent，能够在多个 Agent 间消解不完全知觉问题，协调突发行动[43]。

(2)同时学习问题　若一个多 Agent 系统中包含多个执行不稳定策略且具有学习能力的 Agent，那么每个 Agent 将难以根据自己的行动确定迁移的目的状态。这个问题称为同时学习问题[44]。产生问题的原因是迁移目的状态 s' 未必仅决定于自己的行动 a，而是常常会决定于其它 Agent 的联合行动。

前面提到的 Minimax Q-learning 算法，以及由 Bowling 和 Veloso 提出的算法 WoLF Policy Hill Climbing(PHC)[24] 都是同时学习问题的解决方案之一。若将其他 Agent 看作环境的一部分，在无通信的同时学习机制下，在机器人的协调任务中应用 Q-learning[45]能够得到收敛解。

(3)报酬分配问题　在多 Agent 系统中，即使能够定义整个系统中多个 Agent 联合行动(joint actions)的报酬，通常也难以定义针对各 Agent 个别行动的报酬。这个问题被称为报酬分配问题。典型的例子是机器人足球赛，若有多个 Agent 通过配合成功地完成了射门，射门成功所得的报酬显然不能只分配给最后射门的 Agent，否则除了这个 Agent 以外，其它所有的 Agent 都将不进行学习。因此，要保证系统得到最优的学习结果，研究如何恰当地分配报酬是十分必要的。对报酬分配问题，有代表性的工作有宫崎[46]。

(4)其他重要问题　多 Agent 系统强化学习算法本身还存在着一些需要进一步加以研究的问题：

衡量多 Agent 系统强化学习算法的重要因素之一是它的收敛性[24]，收敛速度的快慢直接影响算法的适用范围和适用程度。今后在这方面的研究工作仍将持续下去。

计算复杂性是采用强化学习算法时所遇到的最为棘手的问题之一，尤其是在机器人足球赛之类对实时性要求严格的场合，强化学习算法往往由于计算速度不能适应要求而导致效率低下。为此，研究人员提出了许多近似算法来逼近最优策略[47-48]。计算复杂性问题渗透在多 Agent 系统强化学习机制的各个研究方向之中，因而具有相当的重要性。

由于强化学习具有无导师的自适应性，因而比较容易和人工神经网络、遗传算法以及其他机器学习算法等相结合。研究证明，人工神经网络在处理状态空间爆炸以及容错性等方面具有优势。Humphrys 将遗传算法应用于报酬分配问题上，取得了较好的实验结果[49]。其他的机器学习方法与强化学习相结合，在状态空间结构化、提高强化学习分析和识别环境的效率等方面也被证明能够起到较好的作用[50]。除此之外，其他一些在传统单 Agent 环境下应用比较广泛的技术，如动态规划技术、规则提取和存储技术等也可以和多 Agent 系统强化学习相结合。因此，和其他技术的结合是多 Agent 系统强化学习算法发展的另一个重要方向。

结语　强化学习算法为多 Agent 系统提供了一种更新行为策略，查找最优解的有效途径。然而，在多 Agent 系统中采用强化学习算法，必然要求多 Agent 系统在通信机制、协商机制等多方面加以变革。本文概述了多 Agent 系统强化学习机制的理论基础、研究进展和较重要的研究课题。从研究历程来看，多 Agent 系统强化学习机制是建立在单 Agent 强化学习算法和对策论基础之上的，已逐渐发展成为一个重要的研究主题。目前，多 Agent 系统强化学习离完全成熟尚有一段距离，还需要通过理论分析和实验验证等手段继续对其机制加以研究。

参 考 文 献

1 Hewitt C. Viewing Control Ctructures as Patterns of Passing Messages. Artificial Intelligence, 1977,8(3):323～364

2 Wooldridge M, Jennings N R. Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge, Jennings, eds. Intelligent Agents, Berlin: Springer-Verlag, 1995. 1～22

3 Wei β G. Learning to Coordinate Actions in Multi-Agent Systems Proceedings of IJCAI'93, 1993

4 Dworman, Garett, Kimbrough S, Laing J. Bargaining by Artificial Agents in Two Coalition Games: A Study in Genetic Programming for Electronic Commerce. In: Proc. of the AAAI Genetic Programming Conf. Stanford, CA, Aug. 1996

5 Kaelbling L P. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, 1996, 4:237～285

6 李宁，高阳，陆鑫，陈世福，等. 一种基于强化学习的 Agent. 计算机研究与发展，2001，38(9)：1051～1056

7 Singh S. Agents and Reinforcement Learning. Miller freeman publish Inc, San Mateo, CA, USA, 1997

8 Bellman R. Dynamic Programming. Prentice-Hall, Englewood Cliffs, NJ, 1957

9 Sutton R S. Learning to predict by the methods of temporal differences. Machine Learning, 1988, 3:9～44

10 Sutton R S. Convergence theory for a new kind of prediction learning. In: Proc. of the 1988 Workshop on Computational Learning Theory, 1988. 421～442

11 Watkins C J C H, Dayan P. Q-Learning. Machine Learning, 8(3): 279～292

12 Tsitsiklis, John N. Asynchronous Stochastic Approximation and Q-learning. Machine Learning, 1994, 16(3): 185～202

13 Rummery G, Niranjan M. On-line Q-learning using connectionist systems: [Tech. Rep. Technical Report CUED/F-INFENG/TR 166]. Cambridge, University Engineering Department, 1994

14 Sutton R S, Barto A G. Reinforcement Learning: Introduction. MIT Press, Cambridge, MA, 1998

15 Sutton R S, Barto A G, Williams R. Reinforcement learning is direct adaptive optimal control. In: Proc. of the American Control Conf. pages 2143-2146. Also published in IEEE Control Systems Magazine, 1991, 12(2): 19～22

16 Tan M. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In: Proc. of Tenth Intl. Conf. on Machine Learning (ML'93), 1993

17 Gu Pan. A Framework for Distributed Reinforcement Learn-ing. In: Proc. of IJCAI95, 1995

18 Littman M L. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In: Proc. of Eleventh Intl. Conf. on Machine Learning (ML'94), 1994

19 Roy N, Pineau J, Thrun S. Spoken Dialogue Management Using Probabilistic Reasoning. In: Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), Hong Kong

20 蔡庆生，张波. 一种基于 Agent 团队的强化学习模型与应用研究. 计算机研究与发展，2001. 9

21 Thomas, Games L C. Theory and application. Halsted Press, 1984

22 高阳，周志华，陈世福，等. 基于 Markov 对策的多 Agent 强化学习模型及算法研究. 计算机研究与发展，2000，37(3)：257～263

23 Barid L C, Moore A W. Gradient decent for general reinforcement learning. In Advances in Neutral Information Processing Systems 11. The MIT Press, 1999

24 Bowling M, Velsoso M. Rational and convergent learning in stochastic games, IJCAI2001, 2001. 1021～1026

25 Hu J, Wellman M P. MultiAgent reinforcement learning: Theoretical framework and an algorithm. In: Proc. of the Fifteenth Intl. Conf. on Machine Learning (ICML-98). Madison, WI, USA, 7, 1998. 242～250

26 Filar J, Vrieze K. Competitive Markov decision process. Springer-Verlag, 1997. Theorem 4.6.4

27 Hu J, Wellman M P. Online learning about other agents in a dynamic multi-agent system. In: Second Intl. Conf. on Autonomous Agents, Minneapolis, 1998. 239～246

28 Bowling M, Velsoso M. Convergence of gradient Dynamics with a variable learning rate. In: Proc. of the Eighteenth Intl. Conf. on Machine Learning, Williams College, June 2001. 27～34

29 Bowling M, Velsoso M. Multiagent learning using a variable learning rate. Artificial Intelligence, 2002

30 Astom K J. Optimal control of Markov decision processes with incomplete state estimation. J. Math. Anal. Appl., 10: 174～205

31 Sondik E J. The optimal control of partially observable Markov decision processes: [Ph. D. thesis]. Stanford University, 1971

32 Singh S P, Jaakkola T, Jordan M I. Model-free reinforcement learning for non-Markovian decision problems. In: Proc. of the 11th Intl. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1994. 288～292

33 Kaelbling L P, Littman M L, Cassandra A R. Planing and acting in partially observable stochastic domains. Artificial Intelligence, 1998, 101: 99～134

34 Smallwood R D, Sondik E J. The optimal control of partially observable Markov decision processes over a finite horizon. Operations Research, 1973, 21: 1071～1088

35 Sondik E J. The optimal control of partially observable Markov decision processes over the infinite horizon: Discounted cost. Operations Research, 1978, 26: 282～304

36 Madani O, Hank S, Condon A. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov Decision Processes. In: Proc. of the 16th National Conf. on Artificial Intelligence (AAAI-99), 1999

37 Hauskrecht M. Planing and control in stochastic domains with imperfect information: [Ph. D. thesis]. Massachusetts Institute of Technology, Cambridge, 2000

38 Littman M L, Cassandra A R, Kaelbling L P. Learning policies for partially observable environments: scaling up. In: Proc. of the 12th Intl. Conf. on Machine Learning, 1995. 362～370

39 Lovejoy W S. A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes. Annals of Operations Research, 1991, 28: 47～66

40 Lovejoy W S. Suboptimal Policies with Bounds for Partially Observed Markov Decision Processes. Annals of Operations Research, 1993, 41: 583～599

41 Simmons R, Koenig S. Probabilistic Robot Navigation in Partially Observable Environments. In: Proc. of Fourteenth Intl. Joint Conf. on Artificial Inteligence, Montreal, Canada, 1995. 1080～1087

42 Thrun S, Fox D, Burgard W. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. Machine Learning, 31: 29～53

43 Arai S, Sycara K. Effective learning approach for planning and scheduling in multi-agent domain. In: Proc. of the 6th Intl. Conf. on Simulation of Adaptive Behavior, 2000. 507～516

44 荒井幸代. 多 Agent 强化学习: 面向实用化的课题、理论和诸技术的结合. 人工智能学会志, 16卷4号, vol7, 2001

45 Matatric J. Reinforcement learning in the multi-robot domain. Autonomous Robots, 1997, 4: 77～83

46 宫崎和光, 荒井幸代, 小林重信. Profit-sharing 在多 Agent 系统强化学习报酬分配中的理论研究. 人工智能学会志, 14卷6号, 1999. 1157～1164

47 Singh S, Jaakkola T, Littman M L, Szepesv'ari C. Convergence results for single-step on-policy reinforcement learning algorithms. Machine Learning, 2000

48 Stone P, Veloso M. MultiAgent systems: A survey from a machine learning perspective. Autonomous Robots, 2000, 8(3)

49 Humphrys M. Action selection methods using reinforcement learning. In: Proc. of the Eighteenth Intl. Conf. on Simulation of Adaptive Behavior, 1996. 135～144

50 Boutilier C. Planning, learning and coordination in multiagent decision processes. In: Proc. of the Sixth Conf. on the Theoretical Aspects of Rationality and Knowledge, 1996. 195～210

(上接第18页)

致谢 感谢联合国大学软件技术研究所对该研究的资助。

参 考 文 献

1 See http://www.w3.org.

2 杨红丽, 郝克刚, 韩俊刚. 基于 Object-Z 的 XPath 形式化语义. 计算机科学, 2003, 30(2)

3 Boag S, et al. XML Path Language (XPath) 2.0. 2002. http://www.w3.org/TR/xpath20/

4 Per Bothner. What is XQuery?

5 Simeon J, Choi B, Fernandez M. The XQuery Formal Semantics: A Foundation for Implementation and Optimization, May 2002

6 Clark J. XSL Transformations (XSLT) Version 2.0, 2002. http://www.w3.org/TR/xslt20/

7 Chamberlin D. XQuery: An XML query language

8 Fernández M, et al. XQuery 1.0 and XPath2.0 Formal Semantics. Nov. 2002. http://www.w3.org/TR/2002/WD_query_semantics_20021115/

9 Duke R, Rose G. Formal Object Oriented Specification Using Object_Z. Macmillan. 2000

10 David C. Fallside. XML Schema part 0: primer, 2001

11 Fankhauser P. XQuery Formal Semantics State and Challenge, 2001

12 Davies J, Wookcock J. Using Z: Specification, Refinement and Proof. Prentice Hall, 1996. http://www.comlab.ox.ac.uk/igdp/usingz

13 Spivey J M. The Z Notation: A Reference Manual. Prentice Hall, 1992

14 Diamond J, et al. Professional XML 2nd Edition. Wrox Press Ltd., 2001

15 Marsh J, et al. XQuery 1.0 and XPath 2.0 Data Model, 15 November 2002. http://www.w3.org/TR/2002/WD-query-datamodel-200211115/

16 Malhotra A, Biron P V. XML Schema Part 2: DAtatypes, 2001. htt://www.w3.org/TR/2001/REC-REC-xmlschema-2-200105-02/

17 Peter. XQuery Formal Semantics State and Challenge. http://ibm.com/developerWorks

18 Fernandez M F, et al. Scott Bodg, Donchamberlin. XQuery 1.0: An XML Query language. 15 November 2002. http://www.w3.org/TR/2002/WD-query-datamodel-200211115/

19 Smith G. The Objice-Z Specification Language. Kluwer Academic Publishers, 1999

20 Thompson H S, Beech D, Maloney M, Mendelsohn N. XML Schema Part 1: Structures, 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010102

21 Sperberg-McQueen C M, Bray T, Paoli J. Extensible Markup Language(XML) 1.0 (Second Edition). 2000. http://www.w3.org/TR/REC-xml/

22 Yang H L, Dong J S, Hao K G, Han J G. Formalizing Semantics of xslt using object-z. In: Proc. of APWeb2003: Asia Pacific Web Conference. Springer-Verlag, April 2003