南京大学

本 科 毕 业 设 计

院　　系　工程管理学院

专　　业　自动化

题　　目　基于多智能体强化学习的
智能仓储移动机器人路径规划

年　　级　2011 级　学 号 111270066

学生姓名　周罗伟

指导老师　杨佩　职 称　讲师

论文提交日期　2015 年 6 月

南京大学本科生毕业论文（设计、作品）中文摘要

题目： 基于多智能体强化学习的智能仓储移动机器人路径规划

工程管理学院 院系 自动化 专业 2011 级本科生姓名：周罗伟

指导教师（姓名、职称）：杨佩，讲师

摘要

针对智能仓储中移动机器人路径规划问题，提出了适用于大规模系统的多智能体强化学习算法(Multi-agent reinforcement learning, MARL)——基于协商的稀疏交互 MARL 算法。算法将传统的均衡型 MARL 算法的均衡思想通过协商的方法引入到稀疏交互中，使得智能体在协调状态处能通过博弈找到均衡联合动作。

本文提出的 MARL 算法由四部分组成：稀疏交互框架，基于协商的均衡动作集合求解，均衡点选取及拓展联合状态的 Q 值迁移。首先，算法采用了单智能体强化学习模型，并且假定每个智能体单独在环境中学习得到最优策略。然后智能体根据瞬时奖励值判断需要协调的状态，在此状态下协商求得纯策略均衡集合，再利用最小方差法选取唯一的联合动作。对于新拓展出来的联合状态 Q 值用环境信息和智能体的协调认知进行初始化。

该算法有两个显著的优点：一方面，与均衡型 MARL 算法相比，新算法的框架为马尔科夫决策模型而非马尔科夫博弈模型，智能体间共享信息少，算法计算复杂度低；另一方面，新算法引入了均衡动作概念，协调避障能力较强。

论文从两方面验证了算法的性能：基于栅格世界基准的测试和基于智能仓储仿真平台的测试。前者检验了新方法的协调性，可延展性及公平性；后者将算法应用于实际系统中并与传统方法 CQ-learning 进行对比，实验结果表明新算法步长数节省约 4.9%，奖励值提升约 1.45%，计算时间节省约 52%。

关键词：多智能体强化学习；协商机制；稀疏交互；知识迁移；路径规划；智能仓储系统

南京大学本科生毕业论文（设计、作品）英文摘要

THESIS: Research on Path Planning Problems in Intelligent Warehouse Systems with Multi-agent Reinforcement Learning

DEPARTMENT: School of Management and Engineering

SPECIALIZATION: Automation

UNDERGRADUATE: Luowei Zhou

MENTOR: Pei Yang, Lecturer

ABSTRACT

This paper presents a new algorithm to solve multi-agent reinforcement learning (MARL) problems, named MARL with Negotiation-based Sparse Interactions (NegoSI). In contrast to traditional sparse-interaction based MARL algorithms, our method incorporated the equilibrium concept, making it possible for agents to select the non-strict Equilibrium Dominating Strategy Profile (non-strict EDSP) or Meta equilibrium for their joint action.

The algorithm consists of four parts: the equilibrium-based framework for sparse interactions, the negotiation for the equilibrium set, the minimum variance method for selecting one joint action and the knowledge transfer of local Q-value. In order to achieve privacy protection, better coordination and lower computational complexity, our method uses three techniques: the unshared value function, the equilibrium solution and sparse interactions.

We evaluated the algorithm in two tests against three criteria: steps for each episode, rewards for each episode and average runtime. The first test with six benchmarks shows fast convergence and high scalability of the algorithm. Then, we compared the NegoSI with CQ-learning on the intelligent warehouse simulation platform. The results of second test showed that, in comparison to the state-of-the-art algorithm, NegoSI is effective in lowering the final steps and increasing the final reward. We also demonstrated our method’s ability to reduce a significant amount of running time compared to the CQ-learning algorithm.

KEY WORDS: multi-agent reinforcement learning; negotiation; sparse interaction; knowledge transfer; path planning; intelligent warehouse systems

目录

第一章 绪论

1.1 研究背景和意义

随着机器人和人工智能理论的不断发展，自主式移动机器人技术日益成熟，并且在工业、军事、医疗、服务等诸多领域得到广泛应用[1-5]。与此同时，机器人所面临的任务也愈加复杂，所处环境由原来的单一机器人、确定性环境转变为多机器人、不确定环境[1,6]。因此，近年来对复杂系统中机器人自主智能控制技术的研究得到了学术界和工业界的广泛关注，而路径规划及导航作为其中的关键性技术成为了目前机器人学的研究热点之一[2,7-8]。

本文以智能仓储中的应用为背景对多机器人路径规划算法展开研究。智能仓库运用大量具有负载能力的智能移动机器人将货架运至工作台，再经工作台的工作人员对各订单货物进行处理，将传统的“人找货物”转变为“货物到人”，有效提高了仓储的运行效率和经济效益[2,8]。不同于传统的自动化仓储系统，智能仓储系统成功地引入了人工智能技术，并综合运用最优化理论、系统决策和博弈论的成果，使得仓储系统的自主协调能力及决策能力很大程度地提升。目前智能仓储系统成功案例有亚马逊公司的子公司 Kiva Systems (见图 1-1)，该公司已将智能仓储技术应用于多个公司的仓储系统管理和供应链实现，如 The Gap 公司和 Staples 公司[8]。

图 1-1 Kiva Systems 智能仓储系统

目前智能仓储系统技术尚处于发展阶段，需要合理解决动态环境下任务分配，多机器人路径规划和协调避障等诸多问题。其中路径规划和协调避障对系统安全

运行尤为重要，需要高度重视。当前智能仓储环境一般通过传统的贪心算法，A*算法[3,8]等方法为机器人规划行进路径，同时采用固定的避障措施避免冲突。然而，这些传统的路径规划及避障方法大多由人为设置，过度依赖于行为控制的程序设计，灵活性差、鲁棒性低，易导致机器人堵塞甚至碰撞问题[9]。针对此类问题，本文提出将以强化学习(Reinforcement learning, RL)为代表的经典机器学习算法对智能仓储路径规划问题进行研究，提出一种具有自主学习能力，环境适应性强并且算法复杂度低的多机器人路径规划算法。

1.2 多智能体路径规划研究现状

路径规划是指在有障碍物的环境中,移动机器人根据一定的评价标准，如路线最短碰撞次数最短等，找到一条从起点到终点的最优或次优的无碰撞路径[10]。目前路径规划技术包括两大类：基于确定环境的全局规划和基于传感探测信息的局部规划。前者是在静态已知的环境中进行路径规划，又称静态路径规划方法，目前应用比较多的方法有：贪心算法，Dijkstra 算法及 A*算法[11]；后者针对环境信息未知的情况，需要根据传感器输入的环境信息实时地进行路径规划，主流的方法有人工势场法，神经网络法，模糊逻辑法等[15-16]。目前，单机器人的路径规划方法已经广泛应用于诸如即时定位与地图构建(SLAM)等实际系统中。

多机器人的路径规划算法主要基于多智能体系统(Multi-agent systems, MAS)的研究。常见的人工智能算法如蚁群算法，模拟退火和遗传算法已在此领域有成熟的应用[7,11]。同时，由单智能体学习衍生而来的多智能体学习(Multi-agent learning)[12-14]也被尝试着用来解决路径规划问题。然而，基于多智能体学习的路径规划技术尚处于理论研究阶段，大部分的研究成果都是基于栅格化地图的仿真实验。其主要原因有两点[1]：（1）目前多机器人路径规划算法鲁棒性和可延展性(scalability)较差，不适用于实际系统；（2）栅格化地图实验能很好地反映算法效果。因此，本文对提出的算法先进行栅格化地图标准(benchmark)的测试，根据多个指标检测算法特性；然后建立实际仓储系统的仿真平台，尝试将算法应用于实际系统中。

1.3 本文研究的主要内容和贡献

1.3.1 本文研究的主要内容

本文对智能仓储机器人路径规划问题的研究建立在多智能体强化学习的理论基础上。而多智能体强化学习理论是对强化学习，博弈论，稀疏交互和知识迁移等理论的综合运用。因此，本文从基本的强化学习理论入手，逐步深入探讨其他相关领域知识，不断完善理论体系，使其能更好地解决智能仓储的实际问题。主要的研究工作包括以下几部分：

（1） 了解智能仓储技术的研究现状，对典型应用系统 Kiva Systems 进行深入分析，建立仿真平台；

（2） 基于强化学习理论以及多智能体强化学习方法，将均衡概念与稀疏交互相结合提出一种基于协商机制的稀疏交互多智能体强化学习算法；

（3） 对传统方法和新方法进行算法实现，利用栅格化地图和智能仓储仿真平台对算法进行测试分析。

1.3.2 本文研究的主要贡献

多智能体强化学习算法(MARL)主要包括两类，一种是在整个联合状态动作空间学习的均衡型 MARL 算法，一种是基于稀疏交互的非均衡型 MARL 算法。前者求解多个智能体的均衡策略，减少碰撞次数，往往能得到较优策略，但是计算速度慢，内存开销很高；后者只在少数场合进行智能体的交互，求解速度快，但是所得策略往往不如前者。本文综合考虑这两类方法的特点，提出了一种既能实现均衡又能稀疏交互的新算法，从一定程度上弥补了这两类方法的不足。

本文主要的创新点如下：

（1） 提出了一种基于协商机制的稀疏交互 MARL 算法，将均衡思想引入到稀疏交互 MARL 算法中。

（2） 基于人类行为学将 MARL 求解过程分解为两部分，第一部分是智能体对环境的认知，体现为对单智能体最优策略的学习，第二部分是智能体对相互协调的认知，体现为智能体间协调配合；

（3） 将多智能体强化学习应用于智能仓储的实际系统中，尝试解决实际问题。

1.4 论文组织结构

本文内容安排如下：

第一章首先介绍了智能仓储的概念及研究现状；其次介绍了多智能体路径规划研究现状。

第二章介绍了强化学习和多智能体强化学习的基础知识，包括马尔科夫决策过程，标准式博弈和马尔科夫博弈。同时也介绍了强化学习领域的其他内容，如稀疏交互和知识迁移。

第三章对论文提出的算法进行详细介绍，包括算法的稀疏交互框架、基于协商的均衡动作集合求解、均衡点选取方法和局部信息的知识迁移等。

第四章对本文所提算法进行对比实验，实验平台包括栅格化地图基准和智能仓储仿真平台两部分。

第五章进行全文总结并提出展望。

本文的知识体系构成如图 1-2 所示：

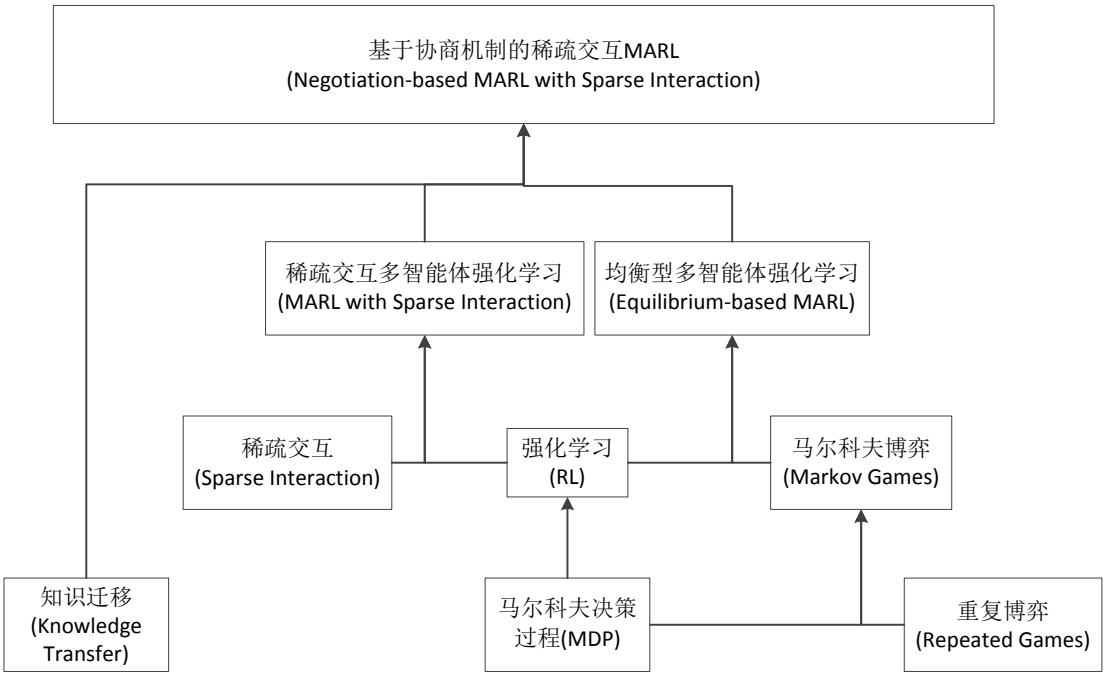


图 1-2 本文知识体系构成

第二章 强化学习与多智能体强化学习

2.1 引言

强化学习利用类似于人类思维中的试错的方法来发现最优策略，其自学习和在线学习的特点使其成为机器学习理论体系中的一个重要组成部分[5]。强化学习求解的问题主要为马尔科夫决策过程(Markov decision process, MDP)，按照方法与模型是否相关可以分为[34]：基于模型的强化学习算法，如 SARSA, Dyna-Q 算法；模型无关的强化学习算法，如瞬时差分算法(TD 算法)和 Q 学习算法。前者通过强化学习先对环境模型进行学习，在此基础上学习最优化的策略；后者直接利用强化学习求解最优策略而忽略环境模型信息。另一方面，依据环境建模不同，强化学习路径规划可以应用于两类情形[35]：离散状态动作 MDP 环境（如迷宫、栅格化智能仓储）和连续状态动作环境（如随机分布障碍物的场地）。前者的研究主要集中在收敛性分析[17]，旨在提高算法的收敛速度，提高找到最优解的成功率[18]；后者的研究主要集中在函数逼近问题，探索与利用的权衡[19]。本文讨论的智能仓储机器人路径规划问题一般被建模为离散状态动作 MDP 问题。多机器人的参与使得智能仓储系统成为多机器人系统，同时路径规划问题为多智能体的 MDP(multi-agent MDP)[20]。解决此问题的主流学习型算法是多机器人强化学习(multi-agent reinforcement learning, MARL)。

多机器人强化学习(MARL)是将强化学习方法运用于多智能体系统，主要有两种途径：独立学习者(Independent learners)和联合动作学习者(Joint action learners)[21]。独立学习者(ILs)中，每个机器人独立使用强化学习完成学习任务，忽略其他机器人，此时机器人的学习环境为动态变化的，算法很难收敛；联合动作学习(JAL)中，每个机器人在自我学习的同时，对其他机器人的动作策略进行学习（或是相互共享），使得多机器人所选策略达到均衡或协调状态（如纳什均衡）。联合动作学习中使用较为广泛的方法是均衡型多机器人强化学习(Equilibrium-based MARL)。均衡型多机器人强化学习以马尔科夫博弈(Markov games)为框架，一般用以解决两类问题[1]：合作式马尔科夫博弈(Team Markov games)及一般式马尔科夫博弈(General Markov games)。本文的研究重心为栅格化智能仓储机器人之间的相互协调，避免发生碰撞或是堵塞，故本文需要解决一般

式马尔科夫博弈下的均衡型多智能体强化学习问题。

Littman 于 1994 年提出了第一种均衡型多智能体强化学习算法：Minimax-Q[12]。在此方法中智能体依据 minimax 的 Q 值选取规则选择相应联合动作。尽管 Minimax-Q 的收敛性较好，但是此方法不满足理性(rationality)[22]，且只适用于解决双智能体零和博弈问题。基于对 Minimax-Q 算法不足之处的分析，Hu 等人提出了解决多智能体一般博弈问题的纳什 Q 学习(NashQ)算法[13]，但是此算法收敛条件很苛刻，并且需要求解算法复杂度很高的混合策略纳什均衡。2001 年，Littman[14]在纳什 Q 学习算法的基础上提出 friend-or-foe Q-learning (FFQ)算法，将交互的智能体分为朋友和敌人分别进行学习，并将纳什均衡分类为敌对均衡和协调均衡，算法收敛性有所改进。这三种方法都是基于混合策略纳什均衡(Mixed Strategy Nash Equilibrium)，其他均衡型多智能体强化学习有:基于相关均衡的 Correlated Q-learning (CE-Q)[23]方法和基于纯策略均衡等多种均衡的协商 Q 学习 (Nego Q-learning)[4]算法等。

本章节我们首先对马尔科夫决策过程模型及典型的强化学习方法做简要介绍，其次将博弈论的概念引入马尔科夫决策过程，得到多智能体强化学习的理论模型——马尔科夫博弈。之后介绍典型的求解马尔科夫博弈问题的算法，纳什 Q 学习和协商 Q 学习。最后简要描述了强化学习的拓展理论，稀疏交互和知识迁移。

2.2 马尔科夫决策过程与强化学习

2.2.1 马尔科夫决策过程

马尔科夫决策过程(Markov decision process, MDP)用以解决连续型决策问题，是强化学习问题的经典模型。MDP 可以用以下定义描述：

定义 2-1 (马尔科夫决策过程) 一个 MDP 可以写成四元组$<S, A, R, T>$，其中S表示状态空间(state space)，A表示动作空间(action space)，$R: S \times A \to \Re$表示奖励函数(reward function)，是从状态动作对到奖励值的一个映射，$T: S \times A \times S \to [0,1]$是状态转移函数(transition function)。

MDP 通过在状态之间引入马尔科夫特性假设降低了强化学习问题建模难度，在这种假设下智能体的下一时刻状态仅取决于此时状态和待执行的动作。该过程

如下图所示：

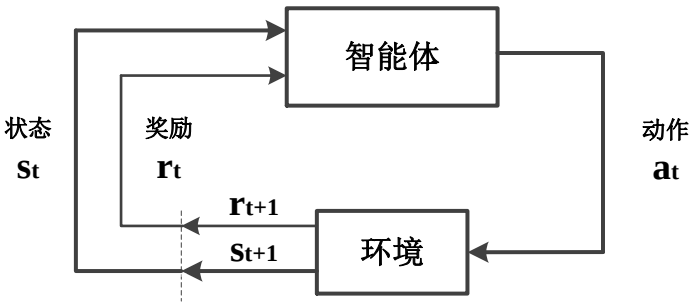


图 2-1 马尔科夫决策过程示意图

在 MDP 问题中，智能体需要寻找一个最优的策略使得其奖励值最大化。其中奖励值可以是平均奖励或是折扣奖励。目前折扣奖励应用较为广泛，故本文采取折扣奖励作为优化目标，即求最优策略 π 满足：

$$V^*(\mathrm{s}) = \max_{\pi} E_{\pi}\{\sum_{k=0}^{\infty}\gamma^k r^{t+k} \mid \mathrm{s}^t = \mathrm{s}\}, \tag{2-1}$$

其中 $V^*(\mathrm{s})$ 表示状态 s 在最优策略下的状态值；策略 $\pi : S \times A \to [0,1]$ 表示在某个特定状态下智能体采取各个动作的概率；E_{π} 是策略 π 下的折扣奖励期望值；t 是任意离散时间步数；k 表示未来时间步数；r^{t+k} 表示在 $(t+k)$ 时间的奖励值；$\gamma \in [0,1]$ 表示折扣因子。这个优化目标同时可以用状态动作对的 Q 值函数来描述：

$$Q^*(\mathrm{s},a) = \mathrm{r}(\mathrm{s},a) + \gamma \sum_{s'} T(\mathrm{s},a,\mathrm{s}') \max_{a'} Q(\mathrm{s}',a'), \tag{2-2}$$

其中 $Q^*(\mathrm{s},a)$ 表示状态动作对 (s,a) 在最优策略下的 Q 值；s' 是下一个状态；$r(\mathrm{s},a)$ 是在状态 s 采取动作 a 的瞬时奖励。

求解 MDP 问题主要有三类方法：动态规划，蒙特卡罗模拟及 TD 算法。动态规划用于已知奖励函数和状态转移函数的场合，利用估计值迭代求解最优解。后两个方法不依赖于环境模型，通过与环境交互估计相应的奖励和转移概率。不同的是，蒙特卡罗每次试验一个策略，当策略执行完成后分配奖励；而 TD 算法每执行一步就能立即更新状态值，算法时间开销较小。在 TD 算法的基础上，Watkins[24]于 1989 年在其博士论文中提出 Q 学习(Q-learning)算法，用 Q 值函数的迭代代替了状态值函数的迭代，取得很显著的效果。此方法已成为当前强化学

习的主流算法。

2.2.2 Q 学习

Q-learning 是一种离策略的 TD 学习，即更新 Q 值函数用到的动作不是下一步要执行的动作。其更新公式为：

$$Q(\mathrm{s},a)\leftarrow(1-\alpha)\,\mathrm{Q}(\mathrm{s},a)+\alpha[\mathrm{r}(\mathrm{s},a)+\gamma\max_{a'}\mathrm{Q}(\mathrm{s}',a')], \tag{2-3}$$

其中 $Q(\mathrm{s},a)$ 表示状态动作对 (s,a) 处的 Q 值；$\alpha\in[0,1]$ 是学习率，影响算法的收敛性。Watkins 证明当学习率满足一定条件且每个状态动作对被访问无限多次时，估计的 Q 值函数 $Q(\mathrm{s},a)$ 一定能收敛到最优的 Q 值函数 $Q^*(\mathrm{s},a)$[36]。Q 学习的一个学习片段(episode)包括以下几个步骤：

（1） 初始化状态及 Q 值函数；

（2） 求解该状态 Q 值最大的动作，按照 $\varepsilon-Greedy$ 动作选择策略选择该动作；

（3） 执行动作后得到下一个状态及智能体将获得的奖励值；

（4） 根据公式(2-3)进行 Q 值更新；

（5） 智能体状态移动到下一个状态，如果此状态是吸收状态或终止状态则一个片段结束；否则返回步骤（2）。

除上述介绍的强化学习算法之外，常用的算法还有 Sarsa 算法，又称为在策略 TD 学习。其与 Q 学习不同之处在于，Sarsa 用来更新状态值的策略就是下一步要执行的策略，而非重新选取。Sutton 在文[34]中还介绍了 TD 算法，Q 学习和 Sarsa 算法的改进算法，如 TD(λ)，Q(λ)和 Sarsa(λ)，主要是对奖励值分配方法进行了改进。以上几种强化学习算法的性能比较详见文献[17]。

2.3 多智能体系统与马尔科夫博弈模型

2.3.1 多智能体系统

2.2 节讨论了单智能体的马尔科夫决策问题，这一节将考虑智能体有多个的情形，研究多智能体间相互协调合作。首先，多智能体系统的概念如下[1,4]：

定义 2-2 (多智能体系统) 一组能自主控制的智能体在共同的环境中相互感知，相互交流，完成预先设定好的任务。这样的系统称为多智能体系统。

目前多智能体系统已经广泛应用于各种场合，如机器人编队，分布式控制，网络连接和数据挖掘等[3,25]。然而多智能体系统的学习(Multi-agent learning, MAL)在现实中的应用较少[1]，实际系统中往往只有单智能体应用的实例。主要原因有两点：相比于单智能体的学习，多智能体所在的环境是动态变化的，各智能体的决策受其他智能体的动作或状态影响，这也导致直接在多智能体系统中使用单智能体强化学习很难收敛；同时，在单一智能体情况下，强化学习一般能收敛到最优策略，然而在多智能体的环境下，智能体之间可能存在冲突，无法求得最优解，取而代之的是较难求解的均衡解（如纳什均衡）。多智能体系统的复杂性决定了问题模型建立的复杂性，为了兼顾系统中各个智能体的利益，协调各个智能体的行为，我们在 MDP 中引入博弈论中的一些概念。

2.3.2 标准式博弈和重复博弈

本小节介绍 MARL 中与博弈论相关的几个重要概念[6]。为了区分强化学习中和博弈论中同被称为“策略”但意义截然不同的两个概念，此节用智能体的动作代替博弈论中策略的概念。

定义 2-3 (标准式博弈，在 MARL 中又称 One-shot 博弈) 标准式博弈可以描述为三元组$<n, A_{1,\cdots,n}, U_{1,\cdots,n}>$，其中$n$表示智能体的数量，$A_k$是智能体$k$的动作集合，$U_k : A_1 \times \cdots \times A_n \to \Re$表示执行联合动作$\vec{a} \in A_1 \times \cdots \times A_n$时，智能体$k$得到的效用值函数，在强化学习问题中表示特定联合状态$\vec{s}$下，联合动作$\vec{a}$对应的 Q 值函数或是状态值函数。

One-shot 博弈概念在求解特定状态下的均衡联合动作时尤为重要。当强化学习算法进行 Q 值函数迭代时，对于同一个状态下的博弈往往需要重复多次以得到稳定的均衡解，这便构成了重复博弈。在重复博弈中引入强化学习得到的迭代更新规则类似(3)，区别在于博弈中不存在状态的概念：

$$Q_i(\vec{a}) \leftarrow (1-\alpha) \mathrm{Q}_i(\vec{a}) + \alpha r_i(\mathrm{t}), \tag{2-4}$$

其中$r_i(\mathrm{t})$表示任意离散时间t执行$\vec{a}$时智能体i能得到的奖励值。

2.3.3 马尔科夫博弈

将重复博弈的概念引入 MDP 中，多智能体在联合状态下选取均衡动作，得到马尔科夫博弈模型：

定义 2-4 (马尔科夫博弈, Markov Games) n 个智能体($n \geq 2$)的马尔科夫博弈可以用五元组<n,$\{S_i\}_{i=1,\cdots,n}$,$\{A_i\}_{i=1,\cdots,n}$,$\{R_i\}_{i=1,\cdots,n}$,T >表示。其中 n 表示系统中智能体数量；$S=\{S_i\}_{i=1,\cdots,n}$ 是各个智能体的状态空间；$A=\{A_i\}_{i=1,\cdots,n}$ 是各个智能体的动作空间；$R_i : S \times A \to \Re$ 是智能体 i 的奖励函数；$T : S \times A \times S \to [0,1]$ 是状态转移函数。 该过程如图 2-2 所示。

如果把智能体 i 的策略记为 $\pi_i = S \times A_i \to [0,1]$，那么所有智能体的联合策略是 $\pi = (\pi_1, \cdots, \pi_n)$。在此策略下智能体 i 的 Q 值函数可以由下式计算：

$$Q_i^{\pi}(\vec{s},\vec{a}) = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_i^{t+k} \mid \vec{s}^{\,t} = \vec{s}, \vec{a}^{\,t} = \vec{a}\}, \tag{2-5}$$

其中 $\vec{s} \in S$ 表示联合状态；$\vec{a} \in A$ 表示联合动作；r_i^{t+k} 是智能体 i 在离散时间($t+k$)的奖励值。和 MDP 中的最优化目标不同的是，马尔科夫博弈一般无法找到最优的策略，取而代之的是均衡的联合策略 π。而这里的均衡策略概念可以转换成智能体在每个状态下选择均衡的联合动作，即进行一次 One-shot 博弈[4]。根据求解该博弈的方法不同可以将 MARL 算法分为多种，如纳什 Q 学习[13]，相关均衡 Q 学习[23]和协商 Q 学习[4]。他们的 Q 值函数更新规则如下所示：

$$Q_i(\vec{s},\vec{a}) \leftarrow (1-\alpha) Q_i(\vec{s},\vec{a}) + \alpha(r_i(\vec{s},\vec{a}) + \gamma \Phi_i(\vec{s}')), \tag{2-6}$$

其中 $\Phi_i(\vec{s}')$ 表示智能体 i 在下一个状态 $\vec{s}'$ 处均衡解对应的期望 Q 值，可以通过在该状态求解 One-shot 博弈得到。

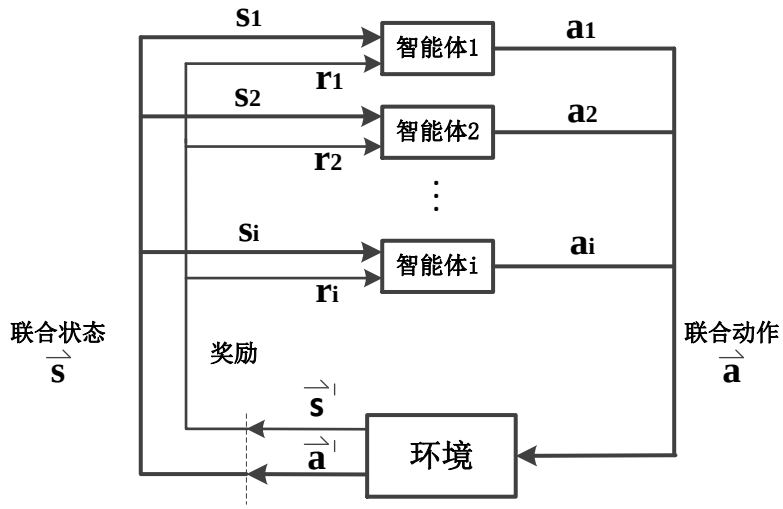


图 2-2 马尔科夫博弈示意图

2.4 典型的多智能体强化学习算法

2.4.1 独立学习者和联合动作学习者

2.3 节简要介绍了马尔科夫博弈模型，基于此模型的 MARL 算法一般称为联合动作学习者(Joint Action Learners)，即所有智能体都是在联合状态动作空间中学习的。除此之外，还有一类 MARL 算法令每个智能体单独利用强化学习进行学习，不进行相互协调，称为独立学习者(Independent Learners)。两者的典型方法如表 2-1 所示。

表 2-1 多智能体强化学习算法分类[6]

		马尔科夫博弈类型	
		合作式马尔科夫博弈	一般式马尔科夫博弈
学习所需信息分类	独立学习者	策略搜索法 策略下降法	MG-ILA (WoLF-) PG LoC 单智能体强化学习 CQ-learning
	联合动作学习者	分布 Q 学习 稀疏网格 Q 学习 Utile 协调	纳什 Q 学习 Friend-or-Foe Q 不对称 Q 学习 联合动作学习者 均衡 Q 学习

独立学习者方法中环境为动态变化，算法收敛性较差，这一点在本文的实验部分有所体现。联合动作学习者中主流的方法是均衡型多智能体强化学习算法，文[4]中给出了这类方法的一般框架，每个片段包括以下几个步骤：

（1） 初始化联合状态及 Q 值函数；

（2） 求解该状态均衡联合动作，按照 $\varepsilon-Greedy$ 动作选择策略选择联合动作；

（3） 执行动作后得到下一个联合状态及各个智能体相应的奖励值；

（4） 根据公式(2-6)进行 Q 值更新，其中 $\Phi_i(\vec{s}')$ 是步骤（2）中获得的均衡值；

（5） 智能体状态移动到下一个状态，如果此状态是吸收状态或终止状态则一个片段结束；否则返回步骤（2）。

2.4.2 节和 2.4.3 节将分别对常见的两种算法均衡型 MARL 算法纳什 Q 学习和协商 Q 学习作详细介绍。

2.4.2 纳什 Q 学习

简单而言，纳什 Q 学习[12]即是在公式(2-6)的 MARL 更新公式中利用纳什均衡替代 $\Phi_i(\vec{s}')$，并在选择动作时(如 $\varepsilon-Greedy$)以较大概率选择纳什均衡动作。此时求解的纳什均衡为混合策略纳什均衡，即满足下述定义[4]：

定义 2-5 (混合策略纳什均衡) 在 n 个智能体的标准式博弈 Γ 中，策略组合(Strategy profile) $\pi^*=(\pi_1^*,\cdots\pi_n^*)$ 是混合策略纳什均衡当且仅当对于任意 $i\in N$ 满足：

$$U_i(\pi^*)\geq\max_{\pi_i'\in\sum_i}U_i(\pi_1^*,\cdots,\pi_{i-1}^*,\pi_i',\pi_{i+1}^*,\cdots,\pi_n^*), \tag{2-7}$$

其中策略 $\pi_i^*:A_i\to[0,1]$ 是关于 A_i 的概率分布；$\sum_i$ 表示智能体 i 的策略空间；对于所有策略组合满足 $U_i(\pi)=\sum_{\vec{a}\in A}\pi(\vec{a})U_i(\vec{a})$。

目前求解混合策略纳什均衡的经典方法有针对两个智能体的 Lemke-Howson 算法和针对任意多个智能体的 Simplicial Subdivision 和 Govindan–Wilson 算法[22,26]。近期 Porter 等人基于循环判断占优策略提出的寻找混合策略纳什均衡的快捷方法[26]，一定程度上提升了纳什 Q 学习算法的求解速度。

2.4.3 协商 Q 学习

传统的均衡型 MARL 方法，如纳什 Q 学习和 FFQ，一般基于求解混合策略纳什均衡。此类方法有四个弊端[4]：各智能体之间的值函数信息必须共享；求解均衡解计算复杂度极高；各智能体必须完全理性(full rationality)；每个状态的 one-shot 博弈中可能有多个均衡点，而每个智能体无法统一到唯一一个均衡点。这些弊端阻碍了均衡型 MARL 方法在大环境多智能体场合下使用。

针对传统方法的这四个问题， Hu 等人于 2014 年提出了基于纯策略均衡的协商 Q 学习算法[4]，有效地解决了这些问题。协商 Q 学习算法在公式(2-6)的 Q 值更新公式中利用纯策略均衡替代$\Phi_i(\vec{s}')$，并在选择动作时(如$\varepsilon-Greedy$)以较大概率选择纯策略均衡解。此处的纯策略均衡由两部分构成：纯策略纳什均衡(PNE)和非严格均衡占优策略组合(non-strict EDSP)。当这两种策略组合都不存在时，用元策略均衡作为纯策略均衡。文献[4]中指出元策略是必定存在的，且上述三种策略都属于对称元均衡。

此方法有诸多优点。首先，各智能体之间不再需要共享数据，如 Q 值函数和奖励函数等，而是通过相互协商得到均衡解。其次，该方法将传统方法求解混合策略均衡的步骤转换成多步求解纯策略均衡的步骤，极大程度地简化了算法的计算复杂度。同时，智能体之间的协商使得各智能体的完全理性不再需要满足，即可以有限理性(bounded rational)。在这种情况下，尽管智能体各自目标会有所冲突，它们仍然会选择彼此合作而不是完全自私。最后，在协商 Q 学习方法中，智能体之间通过协商达成共识，采取相同的联合动作，一定程度上避免了智能体均衡点不统一问题。本文借鉴协商 Q 学习中的协商机制，并通过稀疏交互解决了此方法通信复杂度高的问题。

2.5 多智能体系统中的稀疏交互与知识迁移

在实际应用中，环境的规模，智能体的数量对智能仓储路径规划效率产生很大影响，这就是“维数灾”问题。随着环境规模和智能体数量的增大，路径规划的运算量呈指数增加，智能体长时间无法找到最优解。解决此类方法的有效途径有三种：值函数分解(Value Function Decompostion)[27]，模式转换(Switching Between JAL and IL)[28]和知识迁移(Knowledge Transfer)[20,29-30]。

值函数分解将最优化全局值函数分解为最优化多个局部值函数，典型方法有协调图(CGs)[27]，目前使用较少。模式迁移指在智能体间距离较远时采用独立学习，降低计算复杂度，在智能体间距离较近时采用联合动作学习，达到协调或合作的目标。此类方法以单智能体强化学习为框架，有时称为分布式稀疏交互MDP(Dec-SIMDP)[37]，而非马尔科夫博弈模型。目前 MARL 中提到的稀疏交互方法(Sparse Interaction)主要是指这一类方法，典型算法有 Learning of Coordination[31]和 CQ-learning[28]。知识迁移主要包括均衡迁移(Equilibrium transfer)[20]，值函数迁移(Value function transfer)[30]及协调迁移(Coordination transfer)[29]三类，核心思想是将先前学习数据部分或全部转移到当前学习中，以加快学习速率。

在智能仓储中，大多数情况下机器人相距较远，可以采用单机器人强化学习。当多个机器人聚集时，应当采用联合动作学习方法，对机器人的动作进行协调，避免出现拥堵及碰撞。这便体现了稀疏交互（模式转换）的思想。同时，智能仓储中的每个机器人对环境和如何协调都有自己的先验知识，故当多机器人交互时可以将该知识进行迁移，加速学习过程。下面对多智能体系统中的稀疏交互和知识迁移理论做详细介绍。

在 2.3 节中我们介绍了马尔科夫博弈模型，在这种情况下智能体在所有智能体的联合状态动作空间中学习，即每时每刻都在进行协调交互。但是在现实的多智能体系统中，智能体并不是在所有状态下都需要交互的，也不是所有智能体都需要同时交互。这便体现了稀疏交互的思想。

以文[31]中的二机器人系统为例（如图 2-3 所示）。该环境中有两个房间，连接房间的是一条每次只能通过一个机器人的走廊。两个机器人分别在各自的房间里，他们的目标是到达对方房间的目标点。当机器人在各自的房间行动时，他们不需要考虑其他机器人的状态或是动作，只有到他们同时靠近走廊时才需要与对方协调来顺利通过。这便是多智能体稀疏交互的一个简单示例。

图 2-3 稀疏交互示意图[31]

与基于马尔科夫博弈模型的 MARL 算法不同的是，基于稀疏交互的 MARL 算法以单智能体强化学习为框架，智能体通过学习得出需要和其他智能体协调的状态，在该状态下拓展出其他智能体的联合状态。因此稀疏交互 MARL 算法计算速度与单智能体强化学习速度在同一个数量级，大大提高了很多 MARL 问题的求解效率。典型方法有 Learning of Coordination (LoC) [31]，CQ-learning[28]和 FCQ-learning[32]等。

LoC 算法通过在每个智能体的动作空间中引入伪动作 COORDINATION 确定智能体需要相互协调的状态。CQ-learning 算法中，每个智能体都事先学习好了单独在环境中执行任务的最优策略，然后在进行多智能体学习时通过 Student's t-test 检测立即奖励值是否和先验知识有差别来判断需要相互协调的状态。但此方法中智能体只能看到眼前形势，无法为未来的冲突提前做准备。FCQ-learning 便着力解决这个问题，首先同样用单智能体的最优策略进行初始化，随后在多智能体参与时用 KS-test 检测最先出现 Q 值变化的状态，标记下来作为智能体协调的位置。然而此方法无法适应 Q 值本身的波动。2015 年 AAMAS 国际会议上，Hu 等人提出了 Game abstract 方法[30]。尽管这种方法是基于马尔科夫博弈模型的框架，但是也利用稀疏交互思想简化了运算。文章指出，对于较多机器人的场景，有时同一时间交互的机器人往往只有 2~3 个。这时候就不需要对状态的整个联合动作空间进行 one-shot 博弈了，而只需对参与协调的智能体进行 one-shot 博弈。

前文我们提到各智能体在多智能体交互之前往往自身对环境已有一定认识，比如已经求得独自完成任务的最优策略。此时如果在该环境中使多个智能体进行交互，他们可以将已有的对环境的认识迁移到多智能体的学习中。以文[30]中的

家庭清洁机器人系统为例。起初某户家庭里有一台清洁机器人，它通过一段时间学习对家里的环境已经有一定了解。后来主人发现一台清洁机器人不够用，开始使用第二台清洁机器人，那么这台机器人就不需要重新对家里的环境进行学习，可以通过第一台机器人进行知识迁移得到环境信息。

上述的单智能体最优策略初始化迁移一般称为值函数迁移。常见的知识迁移还有选择性值函数迁移[30]，均衡迁移[20]和协调迁移[29]。选择性值函数迁移的第一步与值函数迁移相同，即使各智能体独立学习计算得到期望 Q 值函数和奖励函数。然后，通过在多智能体系统中用蒙特卡罗方法得到计算实测值，并将预期值和实测值对比，计算每个状态预期值与实际值的“距离”。对于距离小的点使用值函数迁移。均衡迁移作用对象是强化学习算法的片段，判断上一个片段特定状态的均衡动作点是否适用于这个片段的同样状态，即由寻找均衡动作点转变为检验均衡点。如果不符合迁移条件则重新计算该状态均衡动作点。实际上文[20]通过实测证明大约 90%的情况下均衡点都是重复的，故此迁移能很大程度的降低计算量。协调迁移则利用筛选器标记需要协调的联合状态并训练出用于初始化的协调 Q 值函数。文[29]将协调迁移运用于 CQ-learning 中，将拓展出来的联合状态 Q 值用先前训练的协调 Q 值函数初始化。这样相当于将多智能体学习分解为单智能体在环境中的学习和多机器人忽略环境下的协调学习两部分，利用流水线式的计算简化运算难度。

本文提出的方法对协调迁移进行了改进，对拓展联合状态的 Q 值函数进行初始化，这在 3.5 节会进行详细说明。

第三章 基于协商机制的稀疏交互 MARL 算法

3.1 引言

就智能仓储系统而言，参与智能体数目较多，状态空间非常庞大，故传统的 MARL 算法很难解决此类环境下的多智能体路径规划问题。但通过观察我们可以发现，智能仓储中智能体之间需要交互的状态相对于整个状态空间非常少，同时进行交互的智能体数量也极为有限。故本文采用基于稀疏交互的 MARL 算法解决多机器人路径规划问题，提出了一种基于协商机制的稀疏交互 MARL 算法。此方法将基于均衡的 MARL 算法求解均衡策略的特点通过协商机制融入到稀疏交互方法中，充分发挥二者优势，高效便捷地解决了智能仓储多机器人路径规划问题。算法主要由四部分组成：

（1） 稀疏交互框架搭建：对于状态多且智能体数量多的智能仓储系统适合采用基于 MDP 的单智能体强化学习模型，保证每个自主机器人状态和动作的独立性，而并非在整个联合状态动作空间进行学习的马尔科夫博弈模型。但这样又会出现一个问题，智能体选取自身最贪婪的动作，并不顾及其他智能体采取均衡动作。本文提出了一种折中方案，即智能体在需要协调的状态采取类似协商 Q 学习方法中的协商机制进行交互，达到在协调状态处采取均衡动作的效果。

（2） 基于协商的均衡动作集合求解：多个智能体在拓展出来的联合状态协商选择联合动作时一般需要求解均衡解。本文采取了一种协商的方法求解均衡解集合，即每个智能体广播自己倾向的联合动作，与其他智能体协商得到大家都能接受的均衡解集合。

（3） 基于最小方差的均衡点选取方法：求得的均衡解集合中均衡解往往有多个，很难取舍。以往的方法用随机选取方法得到唯一的均衡点[29]，这对于完全竞争的博弈问题适用，但是对于以协调合作为主的智能仓储环境显然不适用。在这种环境下，智能体希望在协调的同时能尽可能地相互合作。本文提出的最小方差方法在保证智能体效用总和较高的情况下（体现合作的思想），选择智能体效用方差最小的均衡解（体现协调公平的思想）。

（4） 拓展联合状态 Q 值迁移：对于拓展出来的联合状态空间 Q 值，早期方法

[28,32]均将其初始化为零。这样不符合现实情况，因为算法假定智能体对当前环境一无所知。近期，Vrancx 等人[29]提出用在空白环境训练的协调 Q 值函数进行初始化。但是这种知识迁移方法只对智能体的协调认知进行迁移而完全忽略了对环境的先验知识。本文提出的方法将这两部分知识融合，保留了智能体对协调和环境两方面的认知。

本章的 3.2 到 3.5 节将以第二章提到的多智能体强化学习理论为基础，结合稀疏交互和知识迁移理论，详细描述新方法的四个组成部分。

3.2 稀疏交互框架

当人们在有限的空间内工作时，他们首先会对如何完成自己的任务有一定的认识，在此基础上再考虑如何与其他人协调工作。受这一人类行为习惯的启发，我们将 MARL 求解过程分解为两部分[33]：第一部分是智能体对环境的认知，即独立在静态环境中学习到自己的最佳策略；第二部分是智能体对相互协调的认知，即智能体比较有其他智能体参与时的工作情况与自己独立工作时的区别（在我们的方法中反映为瞬时奖励值的变化），在这些有差别的状态与其他智能体协调行动。这便是稀疏交互框架的主要思想。

我们首先假定智能体独立在环境中学习得到了最优策略和奖励函数模型。当多智能体同时在环境中工作时，会出现两种情形：如果智能体接收到的状态动作对的瞬时奖励值和他们先前学习到的奖励函数模型完全相同，智能体独立进行动作选择；否则，他们需要在自己的状态动作空间中拓展出需要相互协调的联合状态动作对。具体的学习过程如下（见算法 1）：

（1） 广播得到联合状态。智能体在特定状态选择了一个动作，此时检测到瞬时奖励值与奖励函数模型不相符。那么这个状态动作对就要被标记为“危险”而这个智能体被称为“待协调智能体”。随后，“待协调智能体”广播他的状态动作对信息给其他所有智能体，同时接收其他“待协调智能体”的状态动作对信息。这些状态动作对组成了一个联合状态动作对，记为“协调对”；状态组成了一个联合状态，记为“协调状态”，并加入到“待协调智能体”的状态空间中。

（2） 基于协商的均衡动作求解。当智能体拓展出“协调状态”后就会相互协商选取均衡联合动作来避免冲突。在“协调状态”求解均衡动作包括两个步

骤：①求解均衡动作集合；②在均衡集合中选取唯一的一个均衡点。对于第一步骤，我们借鉴了协商 Q 学习算法[4]中的协商机制。每个“待协调智能体”根据各动作的效用得到自己喜欢的联合动作组合，并广播给其他智能体。这样可以协商求得非严格均衡占优策略组合(Non-strict Equilibrium-Dominating Strategy Profile, non-strict EDSP)，相关算法详见 3.3 节。如果不存在非严格均衡占优策略组合，智能体同样用广播的方法协商出元策略均衡(Meta equilibrium)集合来代替前者。对于第二个步骤，本文提出了一种最小方差法选取最优均衡点，算法见 3.4 节。

（3） 如果既没有检测到瞬时奖励差别，又没有检测到“危险”的状态动作对，智能体独立进行动作选择。

算法 1：基于协商机制的稀疏交互 MARL 算法

输入：智能体 i，状态空间 S_i，动作空间 A_i，学习率 α，折扣率 γ，$\varepsilon-Greedy$ 动作选择策略中的探索因子 ε。

初始化：利用智能体独立学习得到的单智能体最优策略初始化全局 Q 值函数 Q_i。

1: **foreach** 片段 **do**

2: 初始化状态 s_i；

3: **while true do**

4: 从 Q_i 中依据 $\varepsilon-Greedy$ 动作选择策略选择动作 $a_i \in A_i$；

5: **if** (s_i, a_i)被标记为“危险” **then**

6: 智能体广播(s_i, a_i)，同时接收其他“待协调智能体”的联合状态动作差集(s_{-i}, a_{-i})，组成联合状态动作集合($\vec{s}, \vec{a}$)；

/*联合状态差集 s_{-i} 和联合动作差集 a_{-i} 的定义详见 3.5 节*/

7: **if** 联合状态动作($\vec{s}, \vec{a}$)不是“协调对” **then**

8: 标记联合状态动作($\vec{s}, \vec{a}$)为“协调对”，标记联合状态 $\vec{s}$ 为“协调状态”，利用空白环境训练出来的协调 Q 值函数（见公式(3-4)）

初始化联合状态处的局部 Q 值函数 Q_i^J；

9: **End if**

10: 协商得到非严格均衡占优策略组合集（见算法 2）或者是元均衡集合（见算法 3）；

11: 利用最小方差法（见算法 4），从均衡集合中选择出新的联合动作，按照 $\varepsilon-Greedy$ 的原则选择此联合动作为此次要执行的动作；

12: **else if** 检测到了瞬时奖励值与先验知识的偏差 **then**

13: 标记状态动作对(s_i, a_i)为“危险”，智能体广播(s_i, a_i)，同时接收其他“待协调智能体”的联合状态动作差集(s_{-i}, a_{-i})，组成联合状态动作集合$(\vec{s}, \vec{a})$；

14: 标记联合动作$(\vec{s}, \vec{a})$为“协调对”，标记联合状态 $\vec{s}$ 为“协调状态”，利用空白环境训练出来的协调 Q 值函数（见公式(3-4)）初始化联合状态处的局部 Q 值函数 Q_i^J；

15: 依照算法 2，算法 3 和算法 4 协商获得均衡联合动作 $\vec{a}$，按照 $\varepsilon-Greedy$ 的原则选择该联合动作；

16: **end if**

17: 智能体移动到下一个状态 s_i'，得到奖励值 r_i；

18: **if** 联合状态 $\vec{s}$ 是存在的 **then**

19: 更新局部 Q 值，$Q_i^J(\vec{s},\vec{a}) = (1-\alpha)Q_i^J(\vec{s},\vec{a}) + \alpha(r_i + \gamma \max_{a_i'} Q_i(s_i', a_i'))$；

20: **end if**

21: 更新全局 Q 值，$Q_i(s_i, a_i) = (1-\alpha)Q_i(s_i, a_i) + \alpha(r_i + \gamma \max_{a_i'} Q_i(s_i', a_i'))$；

22: $s_i \leftarrow s_i'$；

23: **end while until** 状态 s_i 是吸收状态或终止状态。

3.3 基于协商的均衡动作集合求解

本文的主要贡献之一是在基于稀疏交互的 MARL 的算法中引入了求解均衡动作的思想。这一点使本文的方法明显区别于 CQ-learning [28] 和 Learning of Coordination [31]等传统稀疏交互方法。本节阐述如何通过协商在“协调状态”处得到各智能体的均衡动作集合，包括非严格均衡占优策略组合(Non-strict EDSP)和元均衡集合。非严格均衡占优策略组合的定义如下所示：

定义 3-1 (非严格均衡占优策略组合) 在一个 n 个智能体($n \geq 2$)的标准式博弈 Γ 中，$\vec{e_i} \in A \quad (i = 1,2,...,\mathrm{m})$ 表示纯策略纳什均衡。一个联合动作 $\vec{a} \in A$ 是非严格均衡占优策略组合需满足对 $\forall j \leq n$：

$$U_j(\vec{a}) \geq \min_i U_j(\vec{e_i}), \tag{3-1}$$

寻找非严格均衡占优策略组合的协商算法一般分为两步：第一步是智能体根据每个联合动作的效用选出自己倾向的潜在非严格均衡占优策略组合集合；第二步是各智能体通过协商求出潜在策略集合的交集，即非严格均衡占优策略组合。算法如下所示：

算法 2: 基于协商的非严格均衡占优策略组合求解

输入： 一个标准式博弈 $<n, \{\mathrm{A}_i\}_{i=1,...,n}, \{\mathrm{U}_i\}_{i=1,...,n}>$

/* 注：“待协调智能体” i 只知道 n，$\{\mathrm{A}_i\}_{i=1,...,n}$ 和 U_i */

初始化：“待协调智能体” i 的非严格均衡占优策略组合候选集合为空集 $J_{NS}^i \leftarrow \varnothing$；

“待协调智能体” i 的纯策略纳什均衡最小效用值为正无穷 $MinU_{PNE}^i \leftarrow \infty$；

非严格均衡占优策略组合集为空集 $J_{NS} \leftarrow \varnothing$。

1: **foreach** 联合动作差集 $\vec{a}_{-i} \in A_{-i}$ **do**

/*联合动作差集 a_{-i} 的定义详见 3.5 节*/

2: **if** $\max_{a_i'} U_i(a_i', \vec{a}_{-i}) < MinU_{PNE}^i$ **then**

3: $MinU^{i}_{PNE_{a_i'}} = \max U_i(a_i', \vec{a}_{-i})$；

4: **end if**

5: **foreach** $\vec{a} \in A$ **do**

6: **if** $U_i(\vec{a}) \geq MinU^{i}_{PNE}$ **then**

7: $J^{i}_{NS} \leftarrow J^{i}_{NS} \bigcup \{\vec{a}\}$；

8: **end if**

/* 每个智能体广播自己的非严格均衡占优策略组合候选集合 J^{i}_{NS} 和相应的效用值*/

9: 非严格均衡占优策略组合集合为各“待协调智能体”非严格均衡占优策略组合候选集合的交集 $J_{NS} \leftarrow \bigcap_{i=1}^{n} J^{i}_{NS}$。

然而，在有些情况下纯策略纳什均衡并不存在，因此非严格均衡占优策略组合也不一定存在。在这种情况下，智能体转而求解元均衡点集合。Hu 在文[4]中指出，元均衡也是一种纯策略均衡，并且在标准式博弈中一定是非空的。求解元均衡的充分必要条件[4]如下所示：

定义 3-2 (求解元均衡的充分必要条件) 在 n 个智能体($n \geq 2$)的标准式博弈 Γ 中，联合动作 $\vec{a}$ 是元博弈 $k_1k_2\cdots k_r\Gamma$ 的元均衡的充分必要条件是：

$$U_i(\vec{a}) \geq \min_{\vec{a}_{P_i}} \max_{a_i} \min_{\vec{a}_{S_i}} U_i(\vec{a}_{P_i}, a_i, \vec{a}_{S_i}), \tag{3-2}$$

其中 P_i 是元博弈前缀 $k_1k_2\cdots k_r$ 列在标号 i 之前标号的集合，S_i 是元博弈前缀 $k_1k_2\cdots k_r$ 列在标号 i 之后标号的集合。例如，在一个三个智能体的元博弈 213Γ 中，有 $P_1=\{2\}, S_1=\{3\}; P_2=\varnothing, S_2=\{1,3\}; P_3=\{2,1\}, S_3=\varnothing$。一个元均衡联合动作 $\vec{a}$ 需要满足如下条件：

$$U_1(\vec{a}) \geq \min_{a_2} \max_{a_1} \min_{a_3} U_1(a_1, a_2, a_3)$$
$$U_2(\vec{a}) \geq \max_{a_2} \min_{a_1} \min_{a_3} U_2(a_1, a_2, a_3) \quad (3\text{-}3)$$
$$U_3(\vec{a}) \geq \min_{a_2} \min_{a_1} \max_{a_3} U_3(a_1, a_2, a_3)$$

Hu 等人在文[4]中同样使用一种协商的方法寻找元均衡的集合。在本文的实验中简化了这个过程，直接利用 MATLAB 的 max 函数和 min 函数求得元均衡的阈值。以三智能体系统为例，算法实现如算法 4 所示。Hu 等人还指出非严格均衡占优策略组合和元均衡集合都属于对称元均衡[4]，这在一定程度上保证了本文均衡动作集合求解方法的收敛性。

算法 3: 基于协商的元均衡策略组合求解

输入：一个标准式博弈 $<n, \{A_i\}_{i=1,\ldots,n}, \{U_i\}_{i=1,\ldots,n}>$

/* 注：“待协调智能体”i 只知道 n，$\{A_i\}_{i=1,\ldots,n}$ 和 U_i */

初始化：“待协调智能体”i 的元均衡候选集合为空集 $J^i_{MetaE} \leftarrow \varnothing$；

“待协调智能体”i 的元均衡阈值为正无穷 $MinU^i_{MetaE} \leftarrow \infty$；

初始化元均衡集为空集 $J_{MetaE} \leftarrow \varnothing$。

1: 随机初始化元博弈的标号序列为 $s_1 s_2 s_3$，其集合为 $\{123, 132, 213, 231, 312, 321\}$ ；

2: 调用 max 和 min 函数依据公式(3-2)求出各智能体的元均衡阈值 $MinU^i_{MetaE}$；

3: **foreach** $\vec{a} \in A$ **do**

4: **if** $U_i(\vec{a}) \geq MinU^i_{MetaE}$ **then**

5: $J^i_{MetaE} \leftarrow J^i_{MetaE} \bigcup \{\vec{a}\}$；

6: **end if**

/* 每个智能体广播自己的元均衡候选集合 J^i_{MetaE} 和相应的效用值*/

7: 元均衡集合为各“待协调智能体”元均衡候选集合的交集 $J_{MetaE} \leftarrow \bigcap_{i=1}^{n} J^i_{MetaE}$。

3.4 均衡点选取

在 3.3 节中，我们描述了智能体协商策略组合集合的过程。然而，他们得到的联合动作中往往有多个候选者，难以取舍。在协商 Q 学习方法中，多均衡点的选取是采用一种随机的方法，这样对于完全竞争的智能体有效，但是对于有合作倾向的多智能体系统，该方法仍有很大的改进空间。如在智能仓储系统中，智能体更希望在均衡动作集合中寻找到一个自己最满意的解，同时促进其与其他智能体间的合作又保证公平性。因此，我们提出了一种最小方差的方法来选取合适的均衡点。这个方法首先筛选出所有“待协调智能体”总效用大于特定阈值的联合动作集合，然后在这个集合中找出智能体效用方差最小的解，即最公平的解。选用这个方法出于两个目的：使得智能体在协调的同时尽可能考虑合作并且所选联合动作对每个智能体尽量公平。此处效用和阈值采用了总效用和平均值的二分之一，保证了均衡点一定存在。该方法的详细描述见算法 4。

算法 4: 最小方差法选取均衡点

输入：联合动作集合 $J_{NS}=\{\vec{a}_{ns}^{1},\vec{a}_{ns}^{2},\cdots\vec{a}_{ns}^{m}\}$ 及相应的效用值 $\{U_i^{ns}\}_{i=1,\cdots,n}$

初始化：总效用值的阈值为 τ；

/* 在实验中阈值被设定为每个联合动作总效用值的平均值 $\dfrac{\sum_{j=1}^{m}\sum_{i=1}^{n}U_i^{ns}(\vec{a}_{ns}^{j})}{2m}$

联合动作效用值的最小方差值为正无穷 $MinU_{NS}\leftarrow\infty$；

非严格均衡占优策略组合中的最优联合动作 $J_{BestNS}\leftarrow\varnothing$。

1: **foreach** 联合动作 $\vec{a}_{ns}^{j}\in J_{NS}$ **do**

2: **if** $\sum_{i=1}^{n}U_i^{ns}(\vec{a}_{ns}^{j})<\tau$ **then**

3: $J_{NS}\leftarrow J_{NS}\setminus\{\vec{a}_{ns}^{i}\}$

4: **end if**

5: **foreach** $\vec{a}_{ns}^{j}\in J_{NS}$ **do**

6: **if** $\sqrt{\sum_{i=1}^{n}\sum_{k=i+1}^{n}[U_i^{ns}(\vec{a}_{ns}^{j})-U_k^{ns}(\vec{a}_{ns}^{j})]^2}<MinU_{NS}$ **then**

7: $MinU_{NS} = \sqrt{\sum_{i=1}^{n}\sum_{k=i+1}^{n}[U_i^{ns}(\vec{a}_{ns}^{j}) - U_k^{ns}(\vec{a}_{ns}^{j})]^2}$；

8: $J_{BestNS} = \{\vec{a}_{ns}^{j}\}$；

9: **end if**

经过 3.3 节的均衡动作集合求解和上述的唯一均衡点求解过程，“待协调智能体”确定了唯一一个联合动作。下一步即要根据这个联合动作更新 Q 值函数。这里我们提到的 Q 值函数包括两部分，智能体自身的全局 Q 值函数和“协调状态”处的局部 Q 值函数。这与 LoC 算法[33]中提到更新规则相近。之所以不像其他稀疏交互算法如 CQ-learning 一样只更新局部 Q 值函数，是因为全局 Q 值函数对应的单智能体最优策略可能不完全适用于多智能体系统，甚至对学习算法产生误导。而本文算法对全局 Q 值同样进行学习，会不断校正智能体错误的先验知识，就好比人对环境的认知也会因为其他人的影响而改变。智能体更新完 Q 值函数之后紧接着就转移到下一个状态，并且要选择下一个动作。如果此时智能体独立学习，则智能体只需根据全局 Q 值函数选择动作；如果此时智能体处于“协调状态”，则需要采用 $\varepsilon - Greedy$ 动作选择方法按照局部 Q 值函数选择下一状态的均衡联合动作作为每一个“待协调智能体”的动作。整个算法已经在 3.2 节的算法 1 中给出。

3.5 拓展联合状态 Q 值迁移

在本章前面部分已经指出，多智能体学习已经具备了单智能体学习的先验知识，或者说是迁移知识。有趣的是，我们还可以继续用知识迁移优化我们的新算法。这便是拓展联合状态的 Q 值迁移。在以往的很多文献中[28,31]，“协调状态”刚被拓展出来时，其局部 Q 值总是被初始化为全零。在 2011 年，Vrancx 等人提出了一种“协调迁移”方法，将这些局部 Q 值已用在空白环境训练好的协调 Q 值进行初始化。实验证明这种迁移大大降低了智能体相遇相撞的概率。在现实世界中，这种考虑是完全合理的。多个人在一起完成任务，在发生冲突之前，人们已经对如何协调有了一定的认识。正是基于这种认识，人们通过继续学习（协商）得到特定场景下固定的协调方案。Vrancx 的迁移思想非常有效，然而，不足的

是他仅仅对协调信息进行迁移而完全忽略了环境信息。在现实中表现为人们遇到冲突进行协调时完全忘记了自己本来的任务，这显然不符合实际。同时，我们也在实验中测试发现，将这两种先验知识融合在一起对局部 Q 值函数进行初始化效果很好，也即：

$$Q_i^J(s_i,\vec{s_{-i}},a_i,\vec{a_{-i}})=Q_i(s_i,a_i)+Q_i^{CT}(\vec{s},\vec{a}), \tag{3-4}$$

其中 $Q_i^{CT}(\vec{s},\vec{a})$ 是智能体在空白环境中学到的协调 Q 值函数；$\vec{s_{-i}}$ 称为联合状态差集，表示 $\vec{s}$ 中除去 s_i 的状态集合；$\vec{a_{-i}}$ 称为联合动作差集，表示 $\vec{a}$ 中除去 a_i 的动作集合。下面以5×5的栅格世界为例介绍协调 Q 值函数是如何获得的。在这个栅格环境中，联合动作学习者(JAL)完全忽略环境信息，只学习如何与其他智能体协调。他们被设置好了固定的学习步数，以得到稳定的协调 Q 值模型 $Q_i^{CT}(\vec{s},\vec{a})$。类似文[29]中的处理方法，在 $Q_i^{CT}(\vec{s},\vec{a})$ 中的联合状态 $\vec{s}$ 是用相对行列位置 $(\Delta x,\Delta y)$ 表示的。当智能体企图移动到同一个网格中（如图 3-1 所示），这些智能体获得-10 的惩罚值。其他情况都没有任何奖励。下面以两个智能体的空白环境为例介绍知识迁移过程（见图 3-2）。

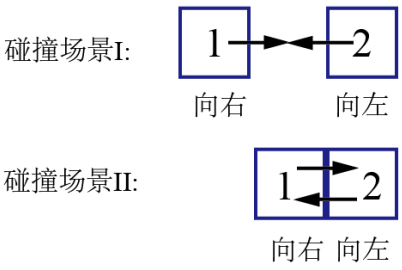


图 3-1 两种空白环境下的冲突实例

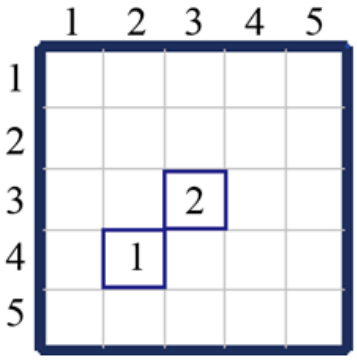


图 3-2 两个智能体的空白环境

智能体 1 和 2 的状态分别是(4,3)和(3,4)，相对状态 $\vec{s}=(x_1-x_2,y_1-y_2)=(1,-1)$。假如有足够的学习步数，智能体学习到他们的 $Q_i^{CT}(\vec{s},\vec{a})$ 为：

$$Q_1^{CT}(\vec{s},\vec{a})=\begin{pmatrix}0&0&-10&0\\0&0&0&0\\0&0&0&0\\0&-10&0&0\end{pmatrix},Q_2^{CT}(\vec{s},\vec{a})=\begin{pmatrix}0&0&-10&0\\0&0&0&0\\0&0&0&0\\0&-10&0&0\end{pmatrix}$$

假定此时智能体处在的“协调状态”如图 3-3 所示，各智能体单独在环境中学得的全局 Q 值函数为：

$$Q_1(s_1,a_1)=(-1,-10,-1,5),Q_2(s_2,a_2)=(-10,-1,5,-1),$$

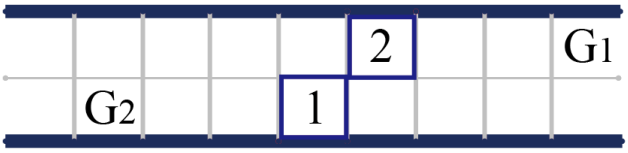


图 3-3 两个智能体的学习环境

那么在此“协调状态”局部 Q 值初始化为：

$$Q_1^J(s_1,\vec{s_{-1}},a_1,\vec{a_{-1}})=Q_1(s_1,a_1)+Q_1^{CT}(\vec{s},\vec{a})=\begin{pmatrix}-1&-1&-11&-1\\-10&-10&-10&-10\\-1&-1&-1&-1\\5&-5&5&5\end{pmatrix},$$

$$Q_2^J(s_2,\vec{s_{-2}},a_2,\vec{a_{-2}})=Q_2(s_2,a_2)+Q_2^{CT}(\vec{s},\vec{a})=\begin{pmatrix}-10&-1&-5&-1\\-10&-1&5&-1\\-10&-1&5&-1\\-10&-11&5&-1\end{pmatrix},$$

可得纯策略纳什均衡为智能体 1 向右走，智能体 2 向左走。如果一开始并未按照此方法初始化，而是按 Vrancx 提出的均衡迁移方法初始化，智能体虽然不会相撞，但是有很大的可能性往回走或是走出界，这种趋势需要学习较多的片段数后才可纠正；如果拓展出的联合状态 Q 值是以全零初始化的，不含任何启发信息先验知识，显然需要花更多时间重新学习。

对于含有三个智能体的“协调状态”，协调 Q 值函数 $Q_i^{CT}(\vec{s},\vec{a})$ 可以类似计算得到，此时的相对状态 $(\Delta x_1,\Delta y_1,\Delta x_2,\Delta y_2)=(x_1-x_2,y_1-y_2,x_2-x_3,y_2-y_3)$ 为四维的而且 Q 值函数是立方型的矩阵。

3.6 本章小结

本章首先在 MARL 模型的选取上放弃了只适用于小规模系统的马尔科夫博弈模型，采用稀疏交互框架（单智能体强化学习）。而在处理智能体协调问题上借鉴了基于马尔科夫博弈模型的协调型 MARL 方法求解均衡动作的思想，巧妙避免了基于稀疏交互 MARL 算法在“协调状态”可能存在的冲突问题。在选取均衡动作时，保留了高效求解纯策略均衡集合的协商方法，同时提出了均衡集合中选取唯一均衡点的方法——最小方差法。本章最后对拓展的联合状态 Q 值初始化进行了初步的讨论，从知识迁移的角度进一步完善了算法。本章所提方法的有效性将在第四章进行验证。

第四章 仿真实验

4.1 智能仓储系统仿真平台的搭建

智能仓储系统主要由三部分组成：工作台（工作人员），货架和移动机器人（如图 4-1 所示）。订单实现过程可以描述为：①智能仓储控制中心接收订单，并且将其拆分为多个单独的任务；②控制中心利用任务分配算法（如拍卖算法）将任务分配给各个机器人，通过无线通信将任务集传给各机器人；③机器人对任务进行初步的路径规划；④机器人完成任务，实现订单。

对于实现订单过程，传统的仓储系统是依靠人或是固定导轨的小车将货物从货架运送到工作台，而智能仓储利用自主路径规划的机器人小车实现了这一功能。在智能仓储系统中，机器人接收到控制中心分配的任务之后先对其进行初步的路径规划，即求得单智能体最优策略。然后机器人将指定货架送至工作台，由工作人员将货物取下，再将货架送回原处。在此过程中机器人之间要考虑协调避障，寻找完成任务的无碰路径。因此在智能仓储中自主机器人路径规划及避障算法是系统实现的重心之一。为了更好地研究路径规划算法，我们建立的智能仓储仿真平台省去了控制中心任务分配及与机器人通信等繁琐的过程，直接使每个机器人完成随机产生的固定数量的任务。

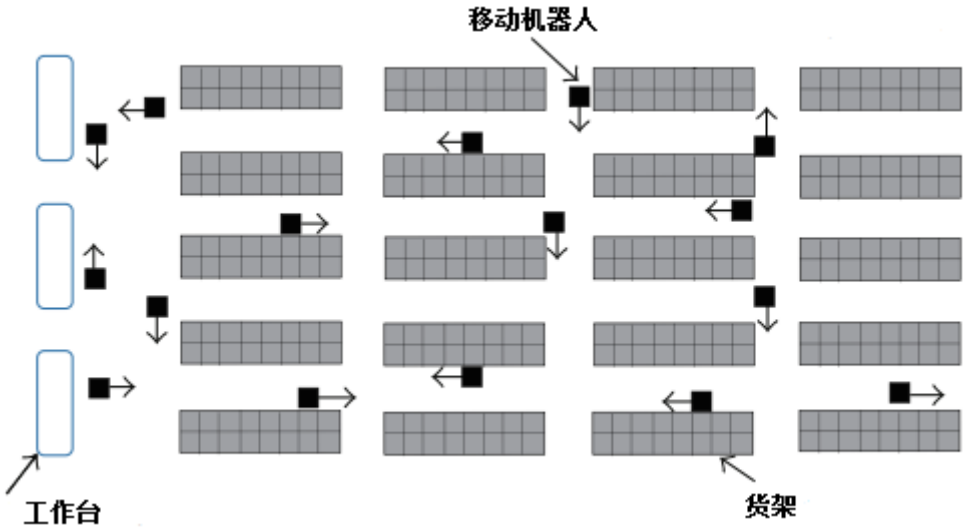


图 4-1 智能仓储系统的三个组成部分

本文仿真平台的所有程序均由 MATLAB 实现，栅格化的仓储平面图如图 4-2 所示。仓储由 21 行 16 列栅格构成，有 4 排 5 列共 20 组货架，每组货架有 6 个子货架，一个 2×4 大小的工作台。栅格的长宽在现实中为 0.5 米，这同时也是机器人和子货架的长宽。仿真仓储的面积为 84 平米，拥有机器人数量为 2~3 个。

在利用智能仓储仿真平台测试算法之前，我们先按照研究惯例，根据算法测试基准(Benchmark)进行测试。

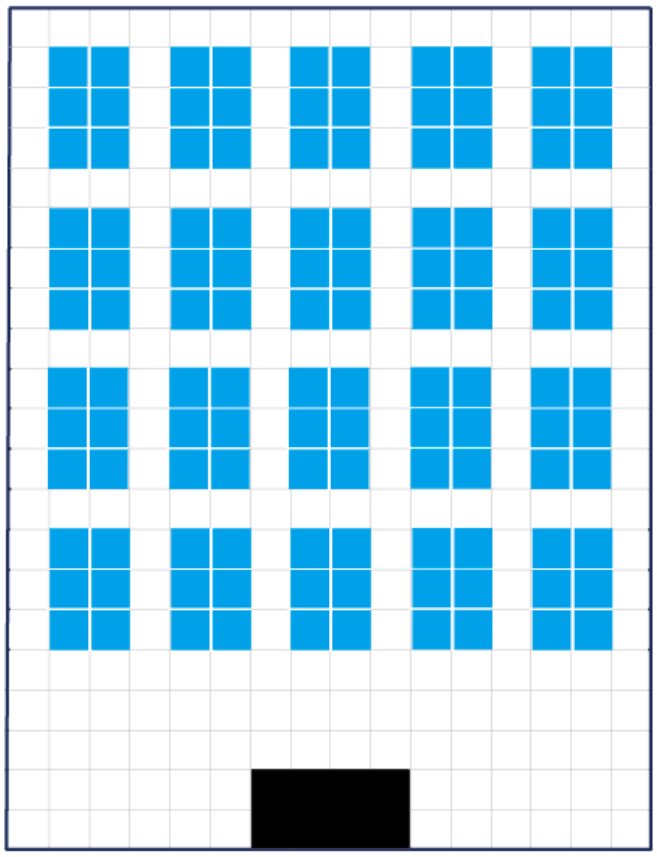


图 4-2 智能仓储仿真平台平面图

4.2 仿真结果与分析

4.2.1 参数设置及评价标准

本节中，我们将本文所提方法与其他经典的 MARL 进行对比。经典方法包括 2010 年提出的 CQ-learning 算法[28], 2015 年提出的值函数迁移协商 Q 学习算法(NegoQ-VFT) [30]和值函数迁移独立学习者方法(ILs-VFT)。本文所提方法在实验中简记为协商稀疏交互算法(NegoSI)。算法测试包括两部分，栅格世界基准的测试和基于智能仓储仿真平台的测试。实验的状态动作及奖励设置如下：

（1） 每个智能体的状态由两部分组成：其所在位置和当前执行的任务编号，故每个智能体的状态空间是位置总数和任务数的组合；

（2） 每个智能体的动作空间包括四种动作：“向上”,“向下”,“向左”,“向右”；

（3） 当智能体到达终点或是任务目标时，他将得到 100 的奖励值。终点是吸收态，也即智能体到达之后就一直处于该状态。当所有智能体都到达终点时，一个片段(episode)结束；

（4） 如果智能体与障碍物或其他机器人碰撞或是出界时，他将得到-10 的惩罚值。同时智能体会反弹回原来的状态；

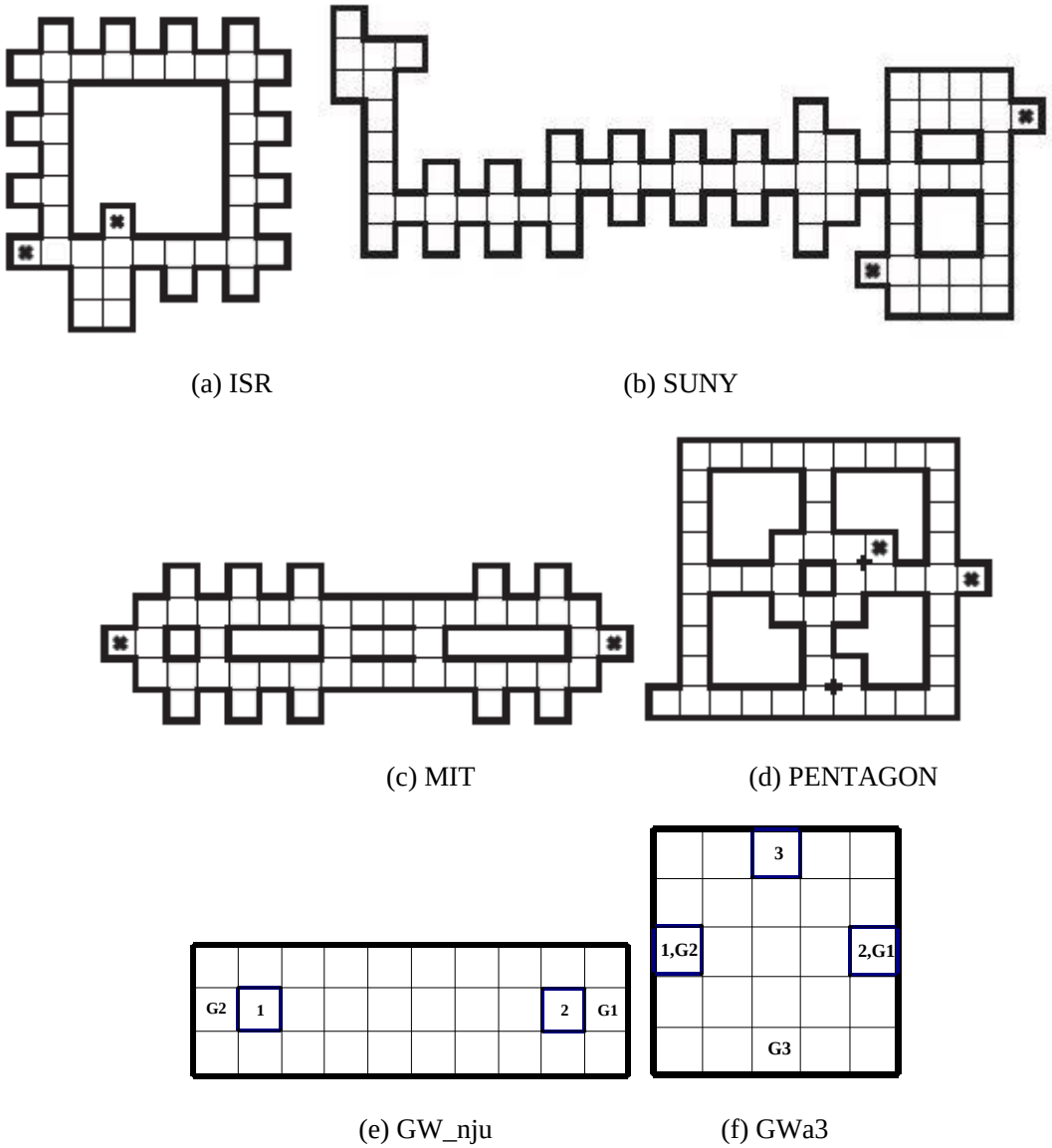
（5） 除（2）（3）两种奖励情况外，智能体每走一步会由于耗电得到-1 的惩罚值。

算法的参数设置如下：学习率 $\alpha = 0.1$，折扣率 $\gamma = 0.9$，探索因子 $\varepsilon = 0.01$。对于栅格世界测试基准，最大片段数和最大步长数均为 2000 和 2000；对于智能仓储系统仿真平台，最大片段数和最大步长数为 8000 和 20000。

算法的评价指标有三个：片段步长数(steps of each episode)，片段奖励值(rewards of each episode)和算法平均运行时间(average runtime)。片段步长数记录了算法在学习过程中每个片段到达终点的步长数，如果未到达终点则记录为最大步长数；片段奖励值记录了算法在学习过程中每个片段得到的奖励值；算法平均运行时间是算法完成学习所消耗的计算机时间。为了作图清晰，本文作图时减少了作图点数，将相邻 100 个数据点取平均值作为新的数据点。所有的结果都进行了 50 次取平均值处理。

4.2.2 基于 benchmark 的算法测试

本文选取的测试基准大部分来源于文[30]和[31]，包括地图 ISR，SUNY，MIT 和 PENTAGON。此外本文提出了两种测试基准，包括 GW_nju 地图和针对三个智能体的 GWa3 地图。基准地图见图 4-3。前四种地图中叉形表示智能体及其终点，后两种地图数字表示智能体，G 表示终点。



(a) ISR　　(b) SUNY

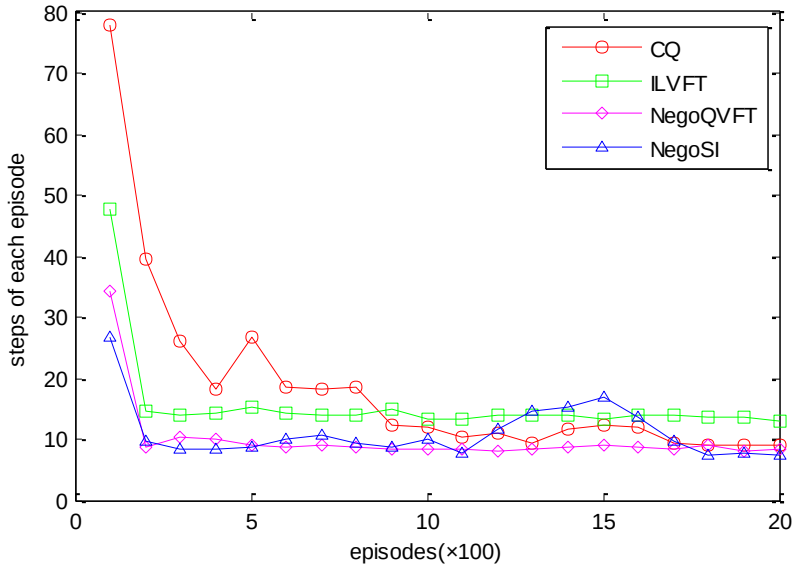
(c) MIT　　(d) PENTAGON

(e) GW_nju　　(f) GWa3

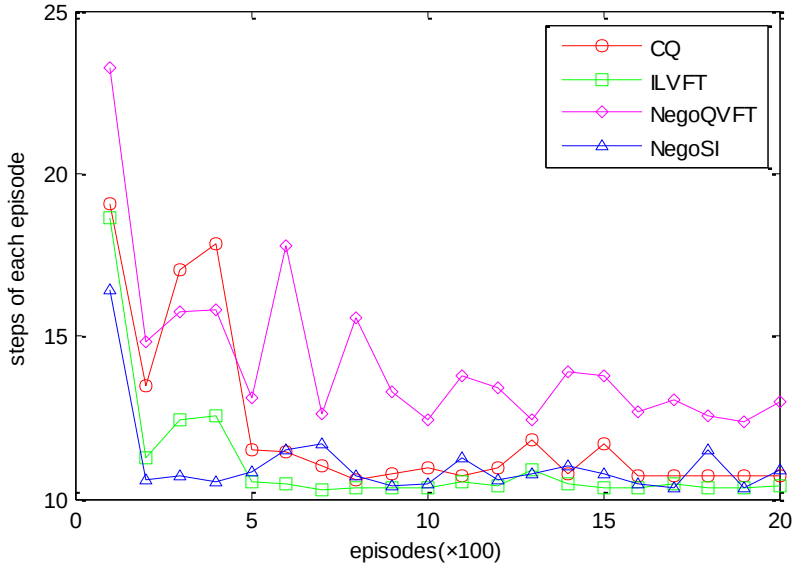
图 4-3 栅格世界测试基准地图

就片段步长数指标而言（见图 4-4），NegoQVFT 算法和 ILVFT 算法步长收敛速度一般非常快（除 SUNY 地图外），我们所提出的 NegoSI 算法收敛性质也较好，CQ-learning 收敛速度最慢。同时 CQ-learning 算法对于有些地图如 SUNY 和

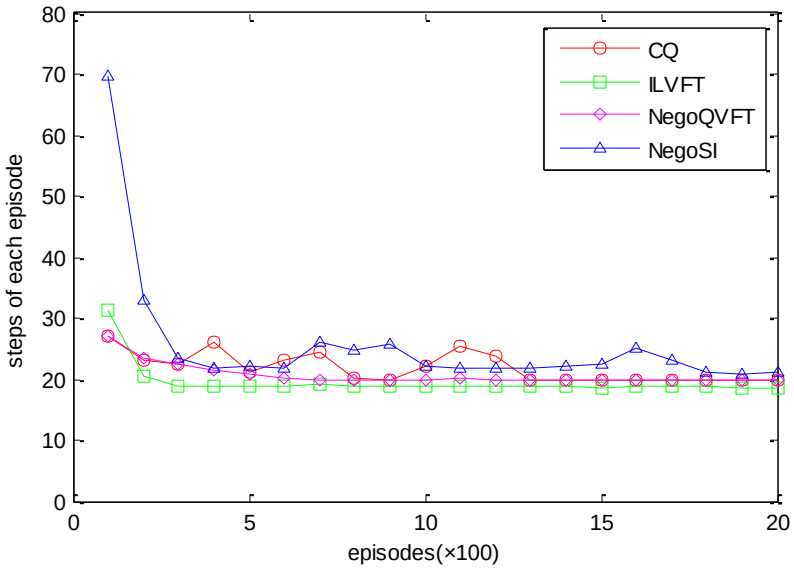
GW_nju 步长振荡非常明显，也就是学习过程中出现冲突，未到达终点的“坏点”数据较多。对于最终片段步长，我们所提出的 NegoSI 算法在 ISR 地图中效果最佳，在其他地图与最短步长相差非常小。CQ-learning 算法最终步长值一般较大；ILVFT 算法最终步长值时好时坏，不太稳定；NegoQVFT 最终步长值除 SUNY 外都非常好，唯一的缺点是该方法在整个联合状态动作空间中学习，消耗内存过大，只适用于栅格世界测试基准这种小型地图，对于大型的智能仓储地图是不适用的。这一点在下一小节中也会详细说明。



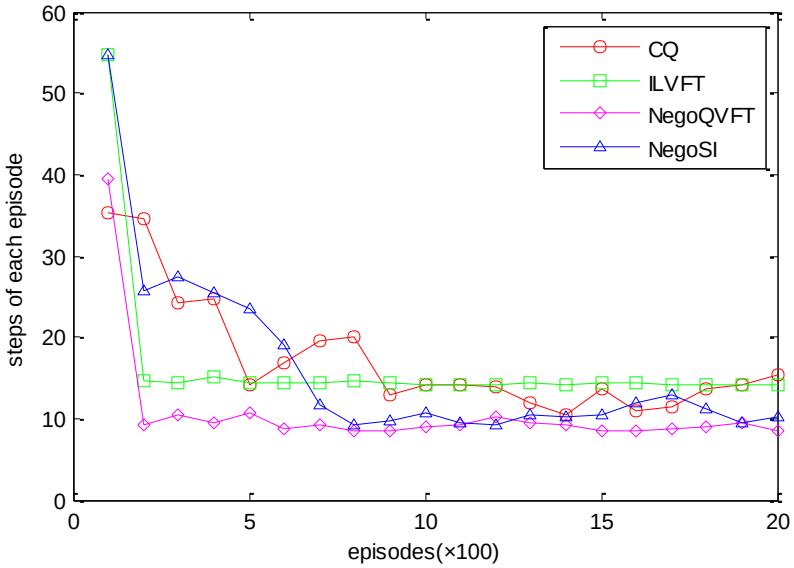
(a) ISR



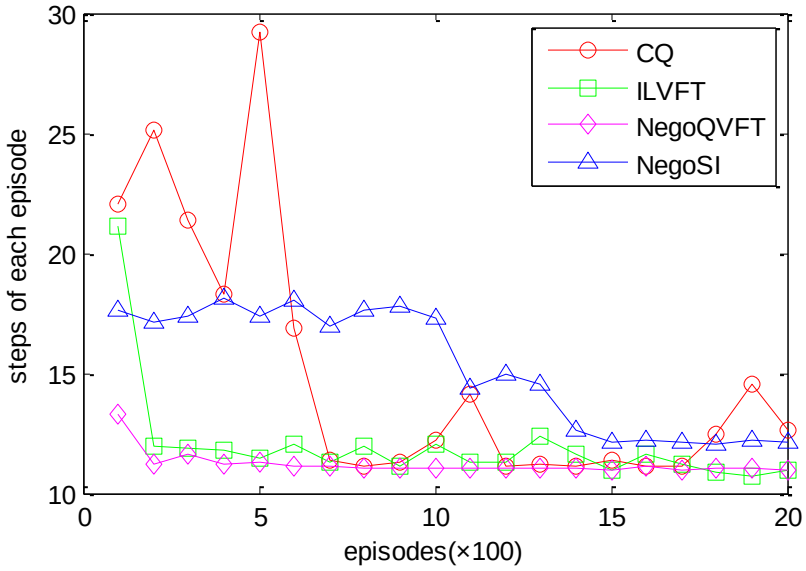
(b) SUNY



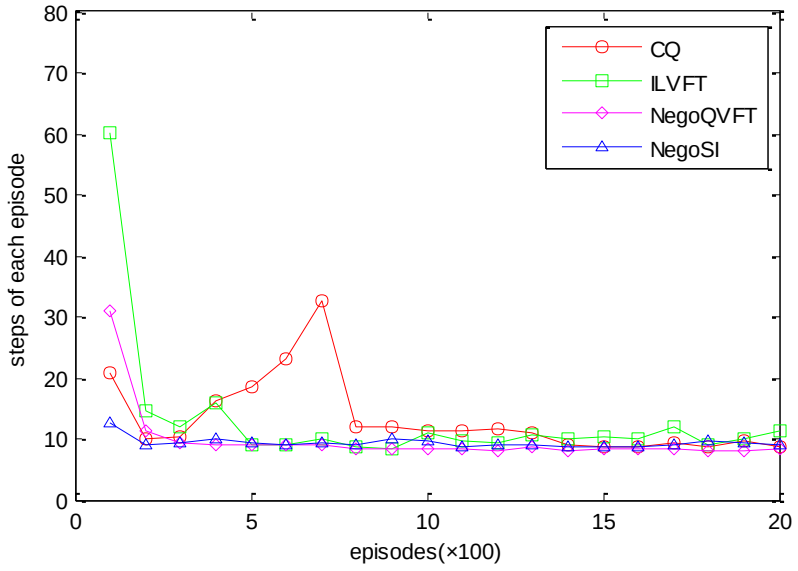
(c) MIT



(d) PENTAGON



(e) GW_nju



(f) GWa3

图 4-4 各地图片段步长数比较

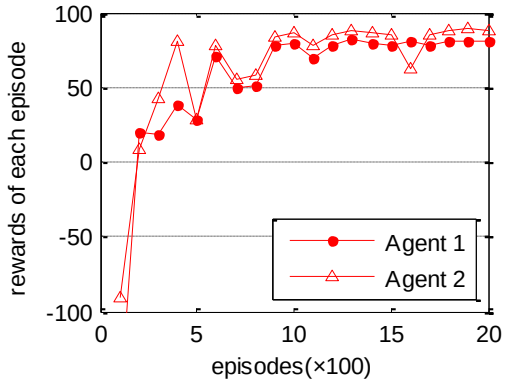
各方法在不同地图下的最终片段步长数如表 4-1 所示:

表 4-1 基准测试地图各算法最终片段步长数

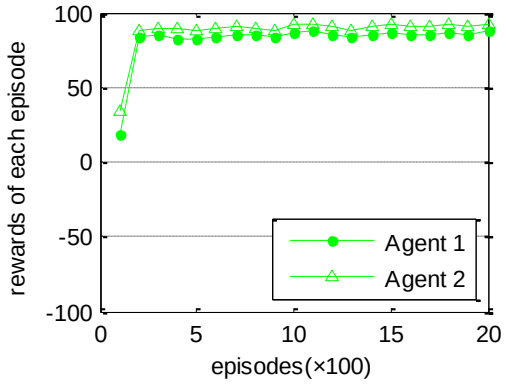
	ISR	SUNY	MIT	PENTAGON	GW_nju	GWa3
CQ	8.91	10.70	19.81	15.32	12.65	8.65
ILVFT	13.11	**10.38**	**18.67**	14.18	**10.94**	11.40
NegoQVFT	8.36	12.98	19.81	**8.55**	10.95	**8.31**
NegoSI	**7.48**	10.92	21.29	10.30	12.11	8.87
最优策略	6	10	18	8	10	8
新方法与最优方法偏差	0	5.14%	14.03%	20.41%	10.62%	6.67%
新方法与CQ算法的偏差	-16.06%	2.00%	7.48%	-32.76%	-4.32%	2.49%

就片段奖励值指标而言（见图 4-5），不同方法在不同地图中效果差别较大，一一描述如下。

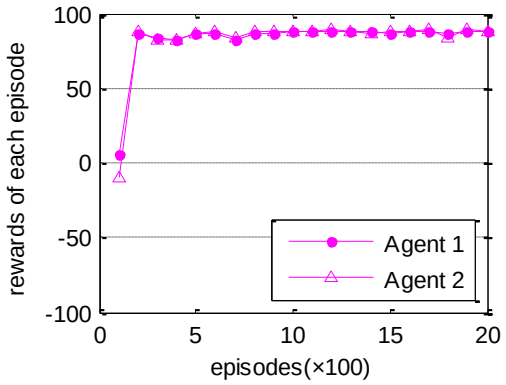
ISR 地图中，我们所提出的 NegoSI 算法的公平性得到很好的体现，即两个智能体奖励值相差非常小。此处的公平性来源于该方法在均衡点选取时采用的最小方差法。相比之下，CQ-learning 和 ILVFT 算法的公平性就稍差一些。同时 NegoSI 算法的片段奖励值在整个学习过程中一直保持着很高的水平，初始片段奖励值是最高的，最终片段奖励值也是最高的。这得益于此方法将均衡思想引入稀疏交互中，降低了由于碰撞产生的奖励值损失。



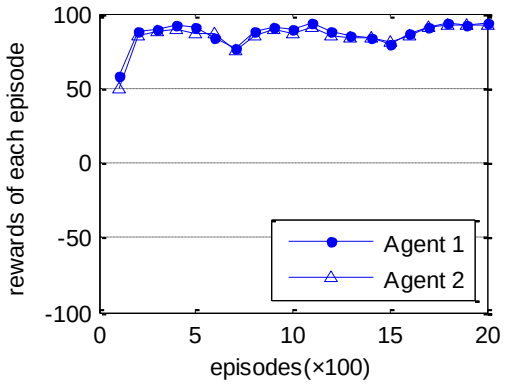
(a) CQ-learning



(b) ILVFT



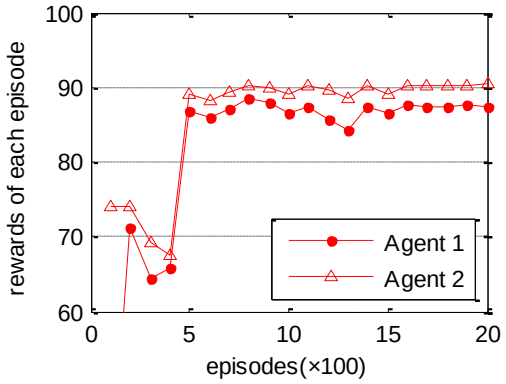
(c) NegoQVFT



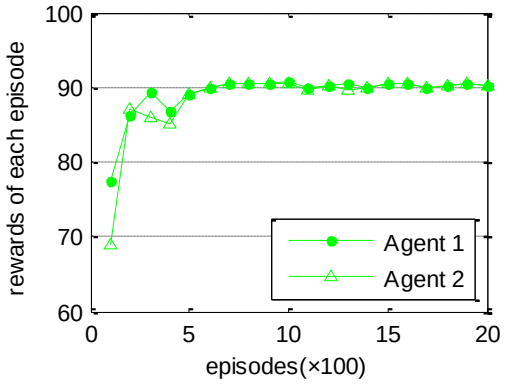
(d) NegoSI

图 4-5(1) ISR 地图各算法片段奖励值比较

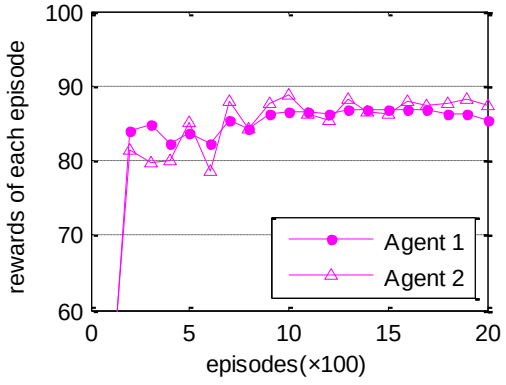
SUNY 地图中，我们所提出的 NegoSI 算法的公平性亦得到很好的体现。此地图中每个智能体有三条最优路径可以到达终点，智能体之间的独立性较强。因此此时基于独立学习者的方法 ILVFT 效果较好，与 NegoSI 算法效果相当。而相比之下，CQ-learning 算法和 NegoQVFT 算法初始奖励值较低，最终奖励值对于不同智能体也有较大区别，缺乏公平性。



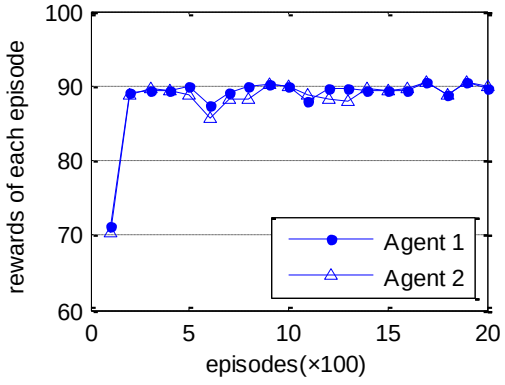
(a) CQ-learning



(b) ILVFT



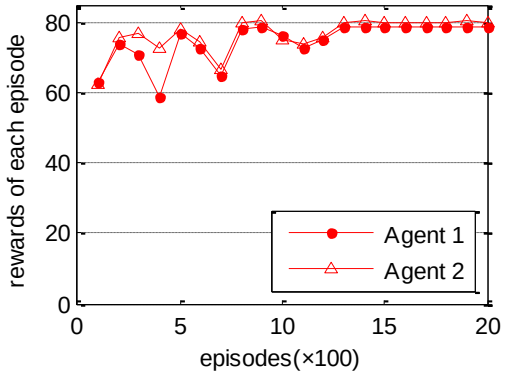
(c) NegoQVFT



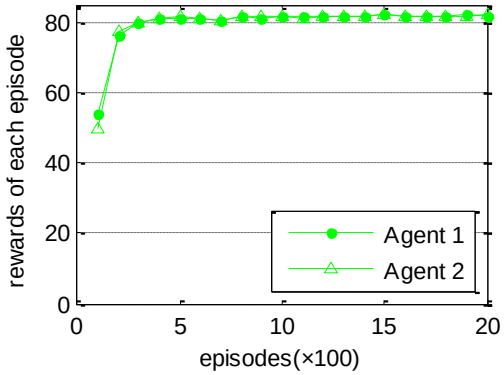
(d) NegoSI

图 4-5(2) SUNY 地图各算法片段奖励值比较

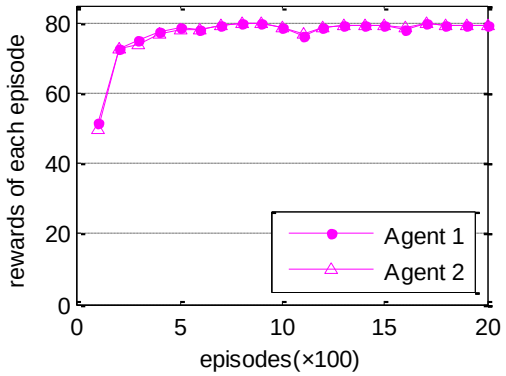
对于 MIT 地图，智能体的可选路径更为丰富，避障方法也较多，四种方法的性能均较好，收敛到了几乎相同的最终奖励值。但在这种情况下，CQ-learning 的奖励值振荡较为明显，收敛特性较差。



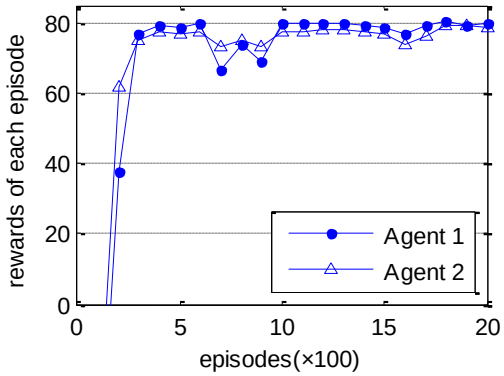
(a) CQ-learning



(b) ILVFT



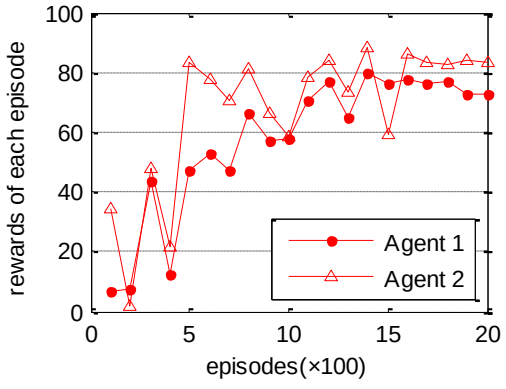
(c) NegoQVFT



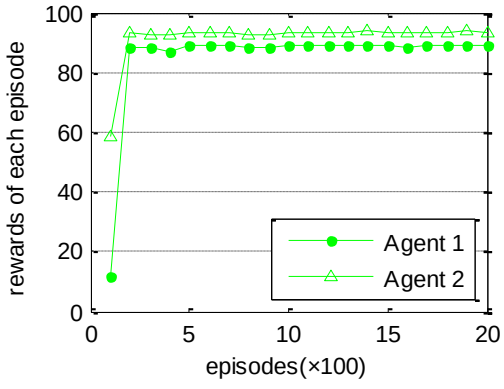
(d) NegoSI

图 4-5(3) MIT 地图各算法片段奖励值比较

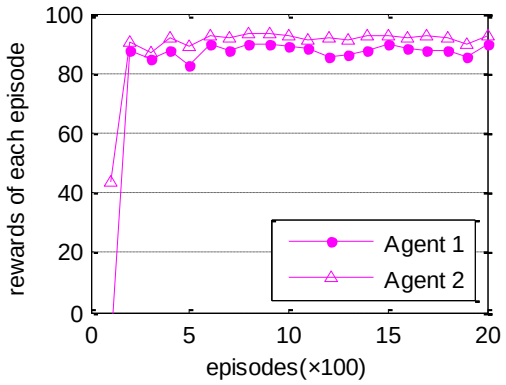
PENTAGON 地图中再次体现了我们所提出的 NegoSI 算法的公平性。此时在四种方法中只有 NegoSI 中每个智能体的奖励值几乎完全相同。其最终奖励值略微逊色于 ILVFT 算法和 NegoQVFT 算法，但仍强于 CQ-learning 算法。



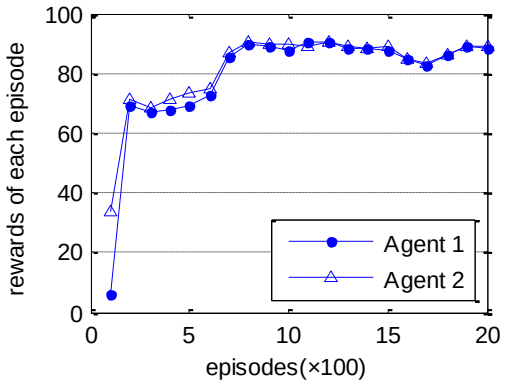
(a) CQ-learning



(b) ILVFT



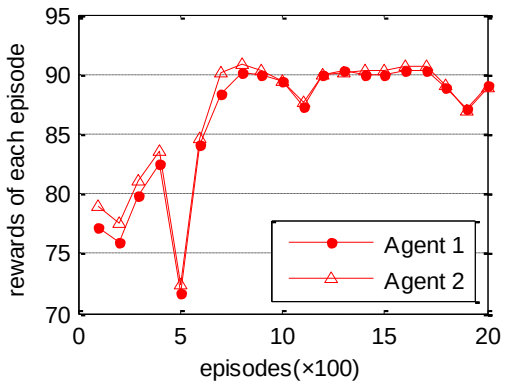
(c) NegoQVFT



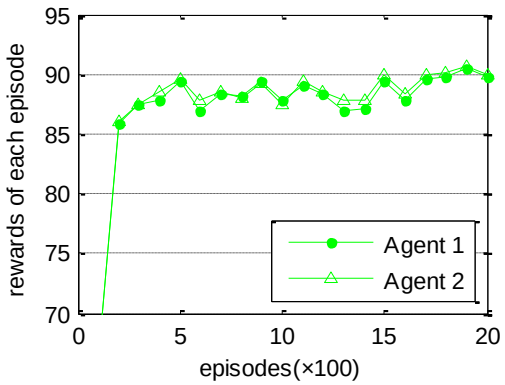
(d) NegoSI

图 4-5(4) PENTAGON 地图各算法片段奖励值比较

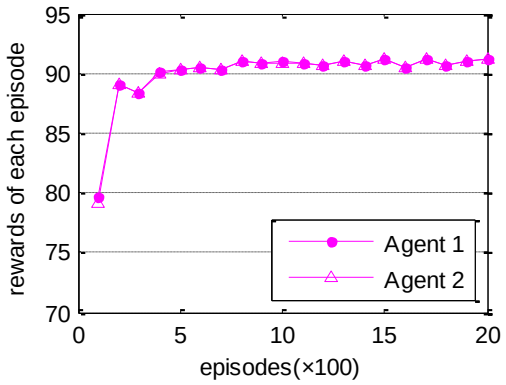
GW_nju 是自行设计的栅格地图。在这个地图中，智能体先前学习的单智能体最优策略一定会发生冲突。即此地图是专门为了检验算法避障能力而设计的。此时智能体最理想的奖励值分别为 93 和 91，即一个智能体按照单智能体最优策略通行，而另一个智能体给其让路。最终结果与理想奖励值接近的为 NegoQVFT 算法和 NegoSI 算法，分别约为 91.39，90.09 和 91.27，91.25。也就是说在 NegoQVFT 算法中，智能体是轮流避让，而在 NegoSI 中智能体学习得到了固定的避让方法。显然后者更安全，更少发生冲突。



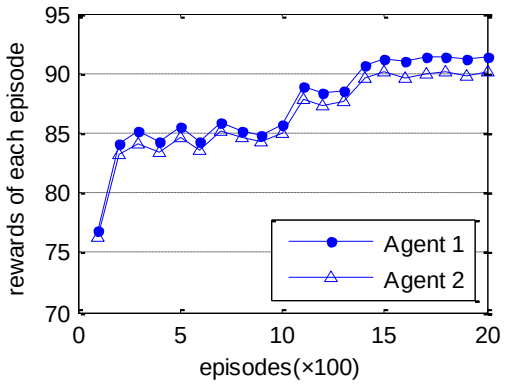
(a) CQ-learning



(b) ILVFT



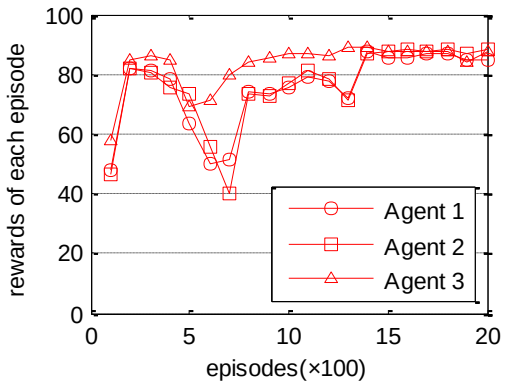
(c) NegoQVFT



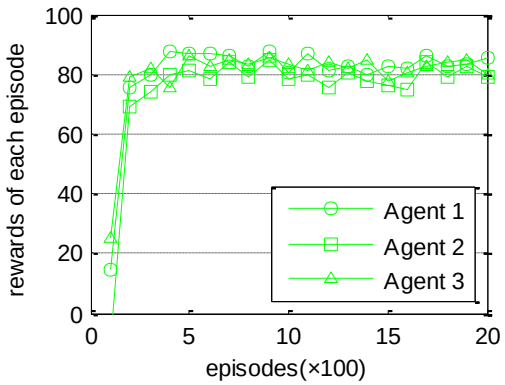
(d) NegoSI

图 4-5(5) GW_nju 地图各算法片段奖励值比较

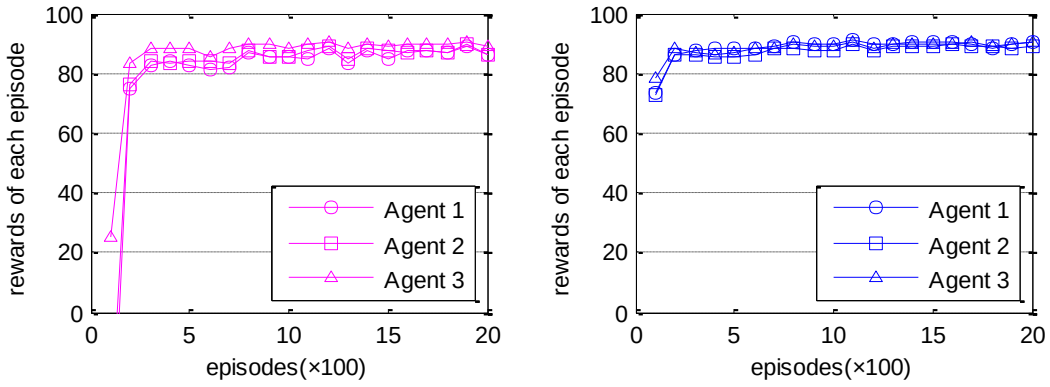
GWa3 是自行设计的三个智能体的栅格地图，此地图对算法的协调避障能力要求也很高。在此种方法中，我们所提出的 NegoSI 算法的奖励值无论是在收敛性还是在绝对数值上都明显优于其他方法，同时智能体间的公平性也得到了很好的保证。这充分体现了 NegoSI 算法对于不同智能体数目场景的可延展性(scalability)。值得注意的是，NegoSI 算法在 GWa3 地图中所用的最终步长数并不是最短的，但是其奖励值却非常高。原因在于新算法鼓励智能体通过增加步数避开碰撞，而传统算法此方面能力较差。



(a) CQ-learning



(b) ILVFT



(c) NegoQVFT　　(d) NegoSI

图 4-5(6) GWa3 地图各算法片段奖励值比较

对于第三个指标平均运行时间，结果如表 4-2 所示。所有测试算法中，运算速度最快的是 ILVFT，一般情况下计算时间是 Q 学习计算单智能体最优策略时间的 5~10 倍；CQ-learning 算法仅仅对“协调状态”处采用联合状态计算，计算量较小，运算速度与 ILVFT 相当；由于 NegoQVFT 算法是在整个联合状态动作空间学习，计算时间非常慢，是 ILVFT 的 5~10 倍； 本文提出的 NegoSI 算法一方面不需要像 NegoQVFT 一样在联合状态动作空间学习，计算量较小，另一方面需要计算“协调状态”处的均衡动作解，计算量稍大，故计算时间介于 CQlearning 算法和 NegoQVFT 算法之间。

表 4-2 基准测试地图各算法平均运行时间

	ISR	SUNY	MIT	PENTAGON	GW_nju	GWa3
CQ	8.54	5.91	13.68	8.52	5.07	5.95
ILVFT	4.91	4.14	9.78	6.56	2.53	7.21
NegoQVFT	13.92	20.18	36.84	18.45	21.89	50.48
NegoSI	16.74	7.33	19.58	16.18	7.08	16.41
Single QL	0.51	0.45	1.29	0.05	0.19	0.07
新方法与 CQ 算法的比例	1.96	1.24	1.43	1.90	1.40	2.76
新方法与 NegoQVFT 算法的比例	1.20	0.36	0.53	0.88	0.32	0.33

4.2.3 基于智能仓储系统仿真平台的算法测试

本小节对移动机器人数量为两个或三个的智能仓储系统进行了仿真测试。在本文的智能仓储仿真平台中，每个机器人有 208 个不同状态，上下左右四个不同的动作。对于两个机器人的仓储系统，机器人 1 的起点终点均为(1,1)，机器人 2 的起点终点均为(1,16)，系统随机产生 60 个任务，每个机器人分配到 30 个；对于三个机器人的仓储系统，机器人 1 的起点终点均为(1,1)，机器人 2 的起点终点均为(1,8)，机器人 3 的起点终点均为(1,16)，系统随机产生 30 个任务，每个机器人分配到 10 个。不同于上一小节对算法基准地图的测试，智能仓储地图测试方法只有两个：NegoSI 算法和 CQ-learning 算法。原因在于 NegoQVFT 算法和 ILVFT 算法不适用于智能仓储系统。

一方面，NegoQVFT 是在整个联合状态动作空间进行学习，内存开销非常大（以三个机器人的仓储为例，NegoQVFT 所需内存为 $208\times208\times208\times10\times10\times10\times4\times4\times4\times8B=4291GB$，而其他方法的内存开销为 $1\sim2MB$），无法在仿真平台上进行测试。另一方面 ILVFT 算法在智能仓储仿真平台中无法收敛。该算法在图 2-3 所示的稀疏交互中无法收敛，而智能仓储环境即是由多个类似图 2-3 的小环境构成的，算法同样不收敛。实际上 ILVFT 算法的收敛性并没有理论保证，出现这种情况实属正常。NegoSI 和 CQ-learning 的测试结果如图 4-6 和图 4-7 所示。

首先分析两个机器人的仓储系统。片段步长数比较见图 4-6(1)，NegoSI 算法的最终步长数为 449.9 步，CQ-learning 算法的最终步长数为 456.9 步，前者比后者节约步数 1.5%。对于片段奖励值（如图 4-6(2)所示），NegoSI 算法一直领先于 CQ-learning 算法，最多时超出 8.8%，最终片段奖励值各机器人各超出 0.67%，0.49%。同时，NegoSI 算法片段奖励值振荡明显小于 CQ-learning 算法，体现了该方法收敛性质良好。运行时间方面，NegoSI 算法平均耗时 2227 秒，而 CQ-learning 算法平均耗时 3606 秒，NegoSI 算法节约了约 38%的计算时间，效果非常显著。

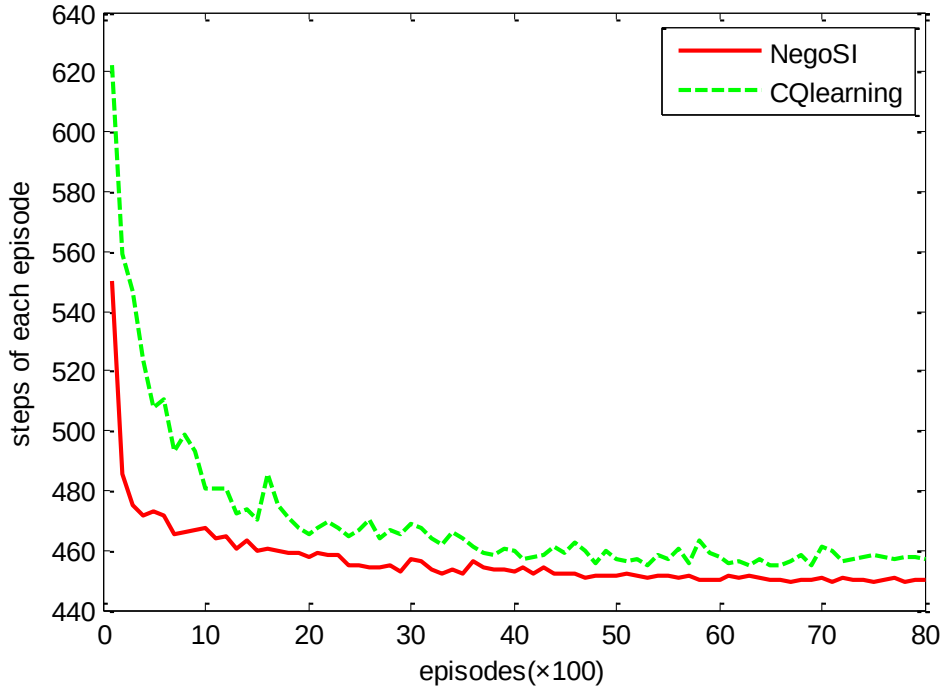


图 4-6(1) 两个智能体时各方法片段步长数比较

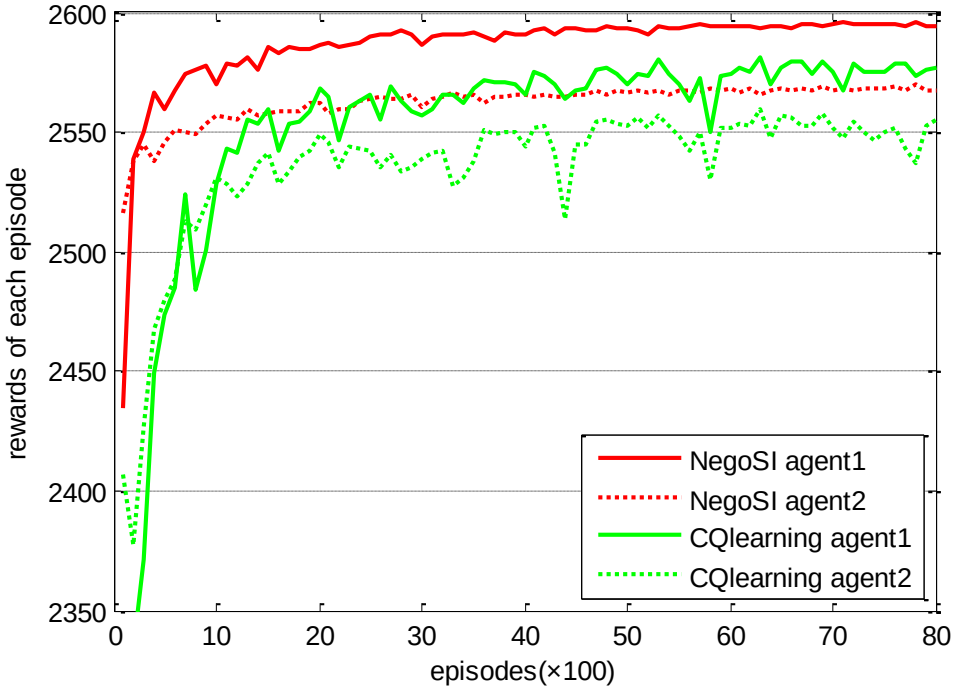


图 4-6(2) 两个智能体时各方法片段奖励值比较

接着我们分析三个机器人的仓储系统。片段步长数比较见图 4-7(1)，NegoSI 算法的最终步长数为 168.7 步，CQ-learning 算法的最终步长数为 177.3 步，前者

比后者节约步数 4.9%。同时由 4-7(1)图可知 NegoSI 算法收敛性能好，稳定性强。对于片段奖励值（如图 4-7(2)所示），NegoSI 算法一直领先于 CQ-learning 算法，最多时超出 16%，最终片段奖励值分别超出 0.82%，0.33%和 1.45%。同时，NegoSI 算法片段奖励值振荡明显小于 CQ-learning 算法，体现了该方法收敛性质良好。运行时间方面，NegoSI 算法平均耗时 1352 秒，而 CQ-learning 算法平均耗时 2814 秒，NegoSI 算法节约了约 52%的计算时间，效果亦非常显著。

注意到 4.2.2 节中，CQ-learning 算法在各个测试地图下计算时间都比 NegoSI 算法计算时间短，但在仓储环境中计算时间却大大超过后者。原因之一是因为 CQ-learning 算法中“协调状态”数多于 NegoSI 算法，在诸如智能仓储的较大环境中，这种差距呈倍数放大。故 CQ-learning 算法中对“协调状态”检索的时间也较 NegoSI 算法显著增加，导致前者计算速率降低。这进一步反映了 NegoSI 算法具有良好的协调能力。

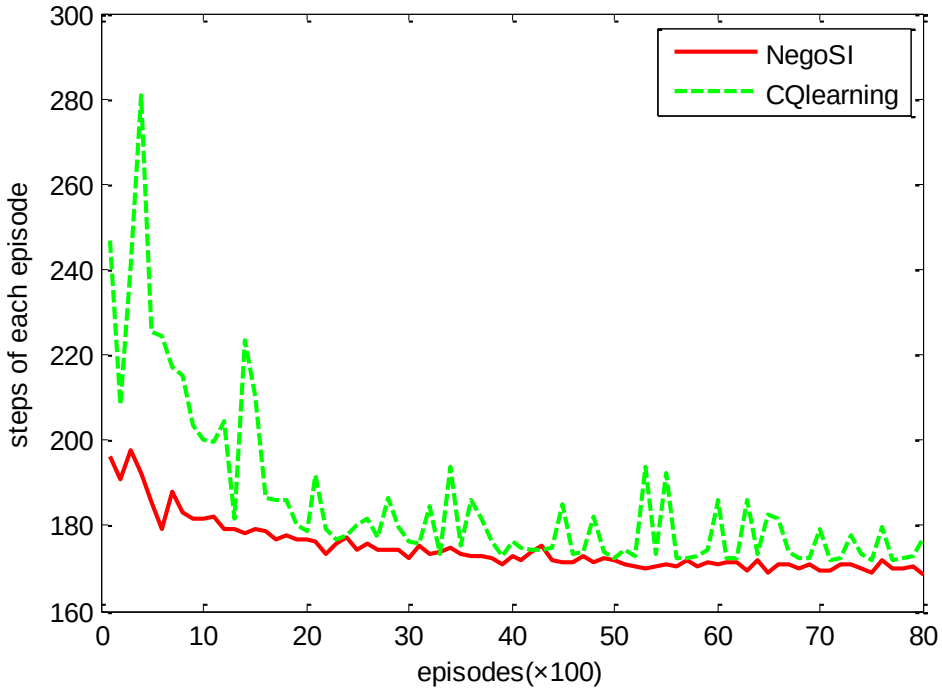


图 4-7(1) 三个智能体时各方法片段步长数比较

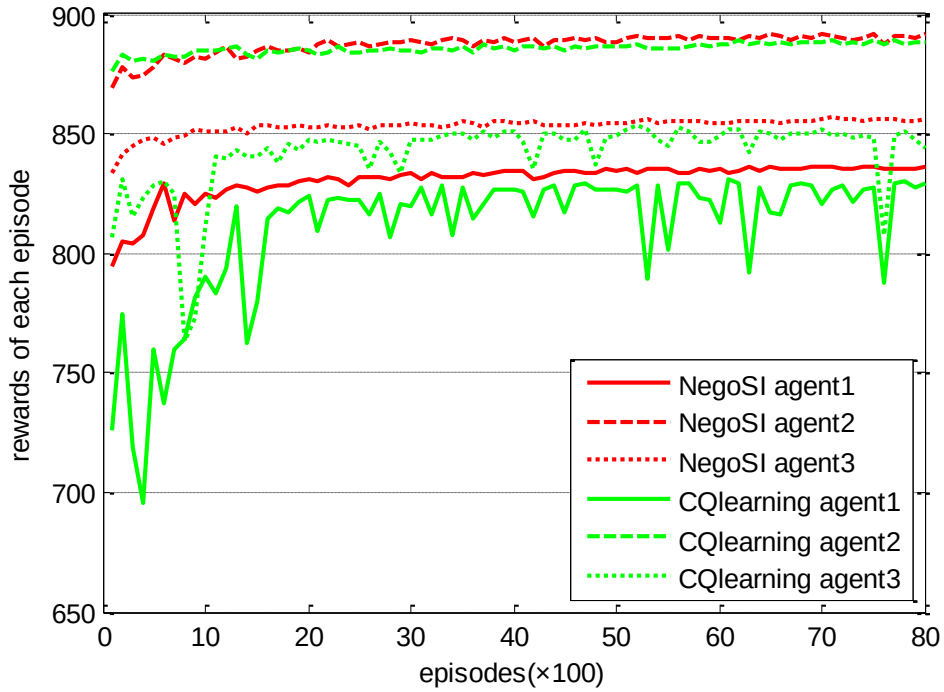


图 4-7(2) 三个智能体时各方法片段奖励值比较

4.3 本章小结

本章主要从两方面测试了本文所提出的基于协商机制的稀疏交互 MARL 算法的效果：基于栅格地图基准的算法测试和基于智能仓储系统仿真平台的算法测试。前者选取了 6 个典型的 MARL 算法测试地图进行测试，清晰直观地反映了所提方法的各项测试指标优劣；后者利用自行编写的智能仓储仿真平台进行测试，将 MARL 算法推广到实际应用系统。两种测试方法都从片段步长数，片段奖励值和算法平均运行时间这三个指标验证了基于协商机制的稀疏交互 MARL 算法的优良特性。

第五章 总结与展望

5.1 本文主要工作总结

本文从多智能体强化学习的角度对智能仓储机器人路径规划问题进行了深入研究。传统意义上的智能仓储机器人路径规划及避障方法大多由人为设置，过度依赖于行为控制的程序设计，灵活性差、鲁棒性低，易导致机器人堵塞甚至碰撞问题。而本文提出的学习型路径规划算法既能辅助各机器人规划合理路径，又能有效地降低机器人碰撞次数，促进机器人间的协调合作。

多智能体强化学习理论建立在强化学习，博弈论，稀疏交互和知识迁移等理论的基础之上，是多门学科交叉结合的产物。故本文在第二章着重介绍了相关理论知识，从马尔科夫决策过程到强化学习，再从马尔科夫博弈到多智能体强化学习，层层深入地搭建了论文相关理论基础。

本文提出的基于协商机制的稀疏交互 MARL 算法是在对现有方法优缺点综合分析的基础上产生的。目前多智能体强化学习算法主要包括两类，一种是在整个联合状态动作空间学习的均衡型 MARL 算法，一种是基于稀疏交互的非均衡型 MARL 算法。前者的均衡性维持了算法的协调能力，避免了智能体碰撞；后者的稀疏交互性质加速了算法计算速度。本文所提算法结合两者优点，利用协商机制将均衡思想引入到稀疏交互方法中，在理论上具有可行性，通过实验也进一步证实了其有效性。

智能仓储系统是本文的研究背景，也是本文所提算法的测试平台。本文开篇详细描述了该领域的研究现状，对典型应用系统 Kiva Systems 进行分析，在此基础上建立了 MATLAB 仿真平台。论文实验部分基于该测试平台对传统方法及提出的新方法进行对比，从片段步长数、片段奖励值和平均运行时间三个角度分析论文所提算法的有效性。

5.2 本文研究的不足之处及未来展望

本文在研究多智能体强化学习算法方面存在以下几个不足之处：

（1） 新算法的理论分析较薄弱，没有对基于非严格均衡占优策略组合及元均衡协商行为的收敛性分析，对最小方差法中阈值选取原则缺乏理论解释；

（2） 新方法在拓展联合状态 Q 值迁移时仅将全局 Q 值信息和协调 Q 值信息简单相加，并未考虑这两者的权重关系，缺乏相关理论解释；

（3） 算法只能判断需要协调的智能体整体，但是不能区分具体哪些智能体之间需要协调，增加了计算量；

（4） 本文对智能仓储系统多机器人路径规划的分析忽略了机器人间通信效率和通信距离的影响，而实际系统中此因素对算法效果有很大影响；

（5） 仅仅考虑了两个或是三个智能体的智能仓储系统，任务数也只有不到一百个，而实际系统中智能体数量可能在十个以上，同时任务数有成千上万个。

针对本文研究的不足之处，未来工作可以围绕两部分展开：（1）对 NegoSI 算法进行深入的理论分析，如收敛性和计算复杂度分析；（2）同时也可以对 NegoSI 如何应用于超大规模的智能仓储进行研究，解决任务分配和机器人通信等问题。

随着当今社会电子商务行业的迅速发展，生产经销商对订单实现的需求也迅速增加。无论是从订单实现的数量要求还是速度要求上看，传统的订单实现仓储已不能胜任此项任务，智能仓储系统的发展迫在眉睫。而大规模智能仓储系统的安全高效运行又离不开优良的学习型多智能体路径规划算法，故 MARL 问题将持续作为多智能体系统研究的热点。

就目前研究状况而言，尽管多智能体强化学习算法的性能不断提升，但与应用到智能仓储等实际系统还有很长一段距离。很多现实问题需要我们解决：多机器人间实时通信，实时定位；动态环境（仓储的位置变化）下的路径规划；机器人数量动态变化对避障性能影响等。除此之外，如何将学习型路径规划算法应用于实时系统，嵌入到现有的成熟方法内也是值得解决的一个问题。总而言之，未来多智能体算法的发展将会更多地联系实际系统，解决实际系统中的各种困难。MARL 算法的研究将对多智能体系统的实际应用产生深刻的影响。

参考文献

[1] L. Buşoniu, R. Babuška and B. D. Schutter. A Comprehensive Survey of Multi-Agent Reinforcement Learning[J]. IEEE Transactions on System, Man, and Cybernetics, Part C: Applications and Reviews, 2008, 38(2): 156-172.

[2] J. Enright and P. R. Wurman. Optimization and Coordinated Autonomy in Mobile Fulfillment Systems[J]. Automated Action Planning for Autonomous Mobile Robots, 2011, 33-38.

[3] L. Zhou, Y. Shi, J. Wang and Pei Yang. A Balanced Heuristic Mechanism for Multirobot Task Allocation of Intelligent Warehouses[J]. Mathematical Problems in Engineering, 2014, Vol. 2014, 10 pages, Article ID 380480.

[4] Y. Hu, Y. Gao and Bo An. Multiagent Reinforcement Learning With Unshared Value Functions[J]. IEEE Transaction on Cybernetics, 2014, 45(4): 647-662.

[5] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, 30(1): 86-100.

[6] A. Nowé, P. Vrancx, and Y. D. Hauwere. Game theory and multi-agent reinforcement learning[M]. Reinforcement Learning, Springer Berlin Heidelberg, 2012, 441-470.

[7] 张亚鸣, 雷小宇, 杨胜跃等. 多机器人路径规划研究方法[J]. 计算机应用研究, 2008, 25(9): 2566-2569.

[8] P. R. Wurman, R. D'Andrea and M. Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses[J]. AI Magazine, 2008, 29(1): 9.

[9] 任建功. 基于强化学习的自主式移动机器人导航控制[D]. 哈尔滨: 哈尔滨工业大学, 2010, 1-7.

[10] 郭娜. 基于模拟退火-Q学习的移动机器人路径规划技术研究[D]. 南京: 南京理工大学, 2009. 1-5.

[11] 王勇. 智能仓库系统多移动机器人路径规划研究[D]. 哈尔滨: 哈尔滨工业大学, 2010. 9-18.

[12] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning[C]. Proceedings of the International Conference on Machine Learning (ICML), 1994, 157: 157-163.

[13] J. Hu and M. P. Wellman. Nash Q-learning for general-sum stochastic games[J]. The Journal of Machine Learning Research, 2003, 1039-1069.

[14] M. L. Littman. Friend-or-foe Q-learning in general-sum games[C]. Proceedings of the International Conference on Machine Learning (ICML), 2001, 322-328.

[15] 宋勇, 李贻斌, 李彩虹. 移动机器人路径规划强化学习的初始化[J]. 控制理论与应用, 2012, 29(12): 1623-1628.

[16] 陆鑫, 高阳, 李宁等. 基于神经网络的强化学习算法研究[J]. 计算机研究与发展, 2002, 39(8): 981-985.

[17] H. H. Viet, P. H. Kyaw and T. Chung. Simulation-based evaluations of reinforcement learning algorithms for autonomous mobile robot path planning[J]. IT Convergence and Services, 2011, 467-476.

[18] T. Kollar and N Roy. Using reinforcement learning to improve exploration trajectories for error minimization[C]. Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2006.

[19] 陈春林. 基于强化学习的移动机器人自主学习及导航控制[D]. 合肥: 中国科学技术大学, 2006, 21-36.

[20] Y. Hu, Y. Gao and B. An. Accelerating Multiagent Reinforcement Learning by Equilibrium Transfer[J]. IEEE Transaction on Cybernetics, 2014, accepted.

[21] C. Claus and C. Boutilier. The dynamics of Reinforcement Learning in Multi-agent Systems[J]. AAAI, 1998, 746-752.

[22] H. M. Schwartz. Multi-agent Machine Learning: A Reinforcement Approach[M]. John Wiley & Sons, 2014.

[23] A. Greenwald, K. Hall and R. Serrano. Correlated Q-learning[C]. Proceedings of the International Conference on Machine Learning (ICML), 2003, 84-89.

[24] C. Watkins. Learning from Delayed Rewards[D]. PhD thesis, University of Cambridge, 1989.

[25] P. Stone and M. Veloso. Multiagent systems: A survey from the machine learning perspective[J]. Autonomous Robots, 2000, 8(3): 345-383.

[26] R. Porter, E. Nudelman and Y. Shoham. Simple search methods for finding a Nash equilibrium[J]. Games and Economic Behavior, 2008, 63(2): 642-662.

[27] J. R. Kok and N. A. Vlassis. Sparse cooperative Q-learning[C]. Proceedings of the International Conference on Machine Learning (ICML), 2004, 61-68.

[28] Y. D. Hauwere, P. Vrancx and A. Nowé. Learning multi-agent state space representations[C]. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2010, 1(1): 715-722.

[29] P. Vrancx, Y. D. Hauwere and A. Nowé. Transfer Learning for Multi-agent Coordination[C]. ICAART (2), 2011, 263-272.

[30] Y. Hu, Y. Gao and B. An. Learning in Multi-agent Systems with Sparse Interactions by Knowledge Transfer and Game Abstraction[C]. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2015, 753-761.

[31] F. S. Melo and M. Veloso. Learning of coordination: Exploiting sparse interactions in multiagent systems[C]. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2009, 773-780.

[32] Y. D. Hauwere, P. Vrancx, and A. Nowé. Solving sparse delayed coordination problems in multi-agent reinforcement learning[J]. Adaptive and Learning Agents, Springer Berlin Heidelberg, 2012, 114-133.

[33] C. Yu, M. Zhang, F. Ren and G. Tan. Multiagent Learning of Coordination in Loosely Coupled Multiagent Systems[J]. IEEE Transaction on Cybernetics, 2015, accepted.

[34] R. S. Sutton and A. G. Barto. Reinforcement Learning: An introduction[M]. MIT press, 1998.

[35] 黄炳强. 强化学习方法及其应用研究[D]. 上海: 上海交通大学, 2007, 15-55.

[36] J. Tsitsiklis. Asynchronous stochastic approximation and Q-learning[J]. Journal of Machine Learning, 1994, 16(3): 185-202.

[37] F. S. Melo and M. Veloso. Decentralized MDPs with sparse interactions[J]. Artificial Intelligence, 2011, 175(11): 1757-1789.

研究成果

1. **Luowei Zhou**, Pei Yang and Chunlin Chen. *Reinforcement Learning in Multi-agent Systems with Negotiation-based Sparse Interactions*. **IEEE Transactions on Cybernetics**. 2015, submitted.
2. **Luowei Zhou**, Yuanyuan Shi, Jiangliu Wang and Pei Yang. *A Balanced Heuristic Mechanism for Multi-robot Task Allocation of Intelligent Warehouses*. **Mathematical Problems in Engineering**, 2014, vol. 2014, 10 pages, Article ID 380480. (SCI indexed)
3. Yuanyuan Shi, **Luowei Zhou**, Jiangliu Wang, Pei Yang and Chunlin Chen. *A Balanced Heuristic Auction Method for Multi-robot Task Allocation of Intelligent Warehouses*. **Proceedings of 2014 Chinese System Simulation Technology & Application**, 2014, 271-276.

致谢

本论文的完结同时也意味着我两年以来智能仓储项目的结束。谨以此文感谢在此期间所有帮助鼓励过我的人。

首先，我要感谢我的毕设导师杨佩老师。从确定思路到开题答辩，到论文结构修改，再到毕业论文完成，整个过程中杨老师始终耐心、负责、严谨地指导我完成工作，并为我的论文提出许多宝贵的意见和建议，使我受益匪浅。在这两年辛苦但是硕果累累的科研项目中，是您耐心地教会我如何阅读文献，如何书写科技论文，让我打下了坚实的科研基础。也是您领导着整个团队，无时不刻鼓舞着我们。

感谢南京大学工程管理学院的所有老师，谢谢你们一直以来对我的敦敦教诲。前三年的专业学习为我的论文打下了坚实的基础。同时，在你们的呵护下，我在大学期间幸福快乐地学习成长，不断提高自己的实力，为未来打下坚实的基础。

同时，感谢窦佳佳学姐和胡裕靖学长。在课题遇到困难的时候，你们总是不厌其烦地为我解答一些专业细节问题，为我耐心地调试程序。你们专心学术的态度深深地影响了我，成为我学习的榜样，也为我的科研学习提供了不竭动力。还要感谢我项目的战友石媛媛同学和王江柳同学，和你们在一起讨论的时光，如项目中一段段的小插曲，令人身心愉悦，同时也给我诸多启发。

我要感谢我的家人。十年寒窗，每当想起你们对我的付出对我的爱，感恩之情便油然而生。在出国留学的这几年里，我定全力以赴，学成归来报答你们的养育之恩，报效祖国。

最后，以此文表达对逝去的 John Forbes Nash 先生最崇高的敬意。