控制与决策
Control and Decision

基于Petri网与多智能体深度强化学习的AGV路径规划

于绍琪, 田玉平

引用本文:
于绍琪, 田玉平. 基于Petri网与多智能体深度强化学习的AGV路径规划[J]. 控制与决策, 2025, 40(5): 1438-1446.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2023.1796

基于 Petri 网与多智能体深度强化学习的 AGV 路径规划

于绍琪[1]，田玉平[2†]

(1. 杭州电子科技大学 自动化学院，杭州 310000；2. 浙大城市学院 信息与电气工程学院，杭州 310015)

摘　要: 在无人仓库系统中, 解决自动导引车 (AGV) 间的碰撞、死锁以及路径规划问题至关重要. 鉴于此, 提出一种用 Petri 网对仓库环境中 AGV 系统进行建模的方法, 以有效解决 AGV 运输货物时产生冲突的问题. 在此基础上, 提出一种多智能体深度强化学习 AGV 路径规划框架, 视 AGV 路径规划问题为部分可观测马尔可夫决策过程, 将深度确定性策略梯度算法扩展至多智能体系统, 通过设计 AGV 的观测空间、状态空间、动作空间以及奖励函数来实现 Petri 网中 AGV 无冲突路径规划. 在设置奖励函数时加入 Petri 网触发条件后的反馈, 以极大程度地减少 AGV 运输货物时拥塞的产生, 增加仓库在规定时间内的送货总量. 此外, 所提出框架将路径分支点设置为智能体, 以有效地应对多个任务起点随机产生以及环境中 AGV 数量时刻变化的情况, 提升神经网络泛化能力. 仿真实验在 AnyLogic 软件平台中进行, 通过对比不同 AGV 规模下的货物运输情况以及奖励函数中有无 Petri 网条件正负反馈的对照实验, 验证所提出路径规划方法的可行性和有效性.

关键词: 多智能体深度强化学习；AGV 路径规划；Petri 网；深度确定性策略梯度算法

中图分类号: TH165$^+$.1；TP23；TP18　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2023.1796

引用格式: 于绍琪, 田玉平. 基于 Petri 网与多智能体深度强化学习的 AGV 路径规划 [J]. 控制与决策, 2025, 40(5): 1438-1446.

AGV path planning based on Petri net and multi-agent deep reinforcement learning

YU Shao-qi[1]，*TIAN Yu-ping*[2†]

(1. School of Automation，Hangzhou Dianzi University，Hangzhou 310000，China；2. School of Information & Electrical Engineering，Hangzhou City University，Hangzhou 310015，China)

Abstract: In unmanned warehouse systems, it is important to solve the collision, deadlock and path planning problems between automated guided vehicles(AGVs). This paper presents a method of a modeling AGV system in warehouse environment with Petri net, which effectively solves the problem of conflict when the AGV transports goods. On this basis, a multi-agent deep reinforcement learning AGV path planning framework is proposed. The AGV path planning problem is regarded as a partially observable Markov decision process, and the deep deterministic policy gradient algorithm is extended to multi-agent systems. Observation space, state space, action space and reward function of the AGV are designed to realize AGV conflict-free path planning in Petri net. Due to the addition of feedback after Petri net trigger condition when setting the reward function, the congestion generated is greatly reduced when the AGV transports goods, and the total amount of delivery in the warehouse is increased within the specified time. In addition, because the proposed framework sets the path branch points as agents, it can effectively cope with the random generation of multiple task starts and the change of the number of AGVs in the environment, and improve the generalization ability of neural networks. Simulation experiments are carried out on the AnyLogic software platform. The feasibility and effectiveness of the path planning method are verified by comparing the cargo transportation situation under different AGV scales and the control experiments with or without Petri net condition positive and negative feedback in the reward function.

Keywords: multi-agent deep reinforcement learning； AGV path planning； Petri net； deep deterministic policy gradient

收稿日期: 2023-12-29；**录用日期:** 2024-10-24.

基金项目: 国家自然科学基金项目 (62073107)；浙江省自然科学基金项目 (LZ21F030002).

†通信作者. E-mail: tianyp@hzcu.edu.cn.

本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

0　引　言

近年来, 自动导引车 (AGV) 因其响应速度快、可控性强、安全性好、效率高, 已作为运输工具广泛应用于半导体制造车间、集装箱码头、生产系统或柔性制造系统 (FMS) 的运输系统. 在运输过程中, 为了提高运输效率, 解决多 AGV 系统的碰撞问题和路径规划问题成为极其重要的课题.

AGV 系统在建模时, 传感器和制动器有限, 存在不可控事件, 因此, 解决车辆碰撞问题是十分重要的. Petri 网作为图形化建模工具, 常用于设计或分析 AGV 系统, 可有效避免 AGV 间的冲突和死锁. 文献 [1] 提出了一种 Petri 网分解的方法来优化 AGV 的路径规划问题, 使用多项式时间内的最短路径算法, 在目标中嵌入惩罚函数, 最终计算出每个子 Petri 网最优路径, 此外, 文中还开发了一种增广的 Petri 网来对多个 AGV 并发动力学进行建模; 文献 [2] 提出了一种 AGV 调度和无冲突路径规划的双目标优化的 Petri 网分解方法, 将 AGV 调度和路径规划问题转化为 Petri 网的双目标最优序列问题, 有效减少了 AGV 运输的总行程时间; 文献 [3] 提出了一种基于 Petri 网的最优控制器方法, 用于防止车辆碰撞, 该方法使用带有标签的变迁来表示不可区分事件和不可控事件, 从而解决了由这些事件引起的高计算复杂性, 并使得控制程序的开发更加便利; 文献 [4] 将 AGV 系统及其环境建模为库所延时 Petri 网, 提出了一种设计控制结构的方法, 避免了 AGV 冲突并减少了死锁. 上述文献用 Petri 网建模解决了 AGV 的冲突问题并解决了稳态环境下的静态路径规划问题, 但是没有考虑动态路径规划下 Petri 网造成的时间成本以及拥塞问题.

在现实世界中, AGV 经常在动态变化的环境中进行运输工作, 需要应对各种各样的情况, 因此, 在进行动态路径规划时需要具备灵活性和适应性. 文献 [5] 提出了一种时间窗口和执行窗口重叠的方法测试候选路径以获得一组可执行的路径, 但是, 在路径规划时没有考虑安全性问题; 文献 [6] 提出了一种基于 RRT*的动态路径规划算法, 当出现未知的移动障碍物时, 可及时修改 AGV 的路径, 但是实验时环境中只设置了少量移动障碍物, 在大量移动障碍物下, 算法的性能还需要进一步实验研究. 强化学习算法通过不断试错来获得奖励值, 以此优化智能体的决策, 它不依赖环境, 因此, 成为路径规划研究的热点方法之一. 多智能体强化学习根据通信可划分为 3 类: 1) 完全无通信算法, 如 IQL[7] (independent Q-learning), 此类算法由于在学习过程中不进行通信, 可能会导致每个智能体根据自身观测很难学习到合理的策略; 文献 [8] 提出了一种基于强化学习的 QLBWR(λ)($Q(\lambda)$ learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs) 算法来解决动态路径规划问题, 将节点作为智能体, 用 Boltzmann softmax 策略增加智能体探索可能性, 在奖励函数中加入了行驶时间和曼哈顿距离, 实验结果表明, 算法有效减少了系统拥塞, 但是文中没有考虑 AGV 的安全性问题; 文献 [9] 提出了一种结合深度Q学习和卷积神经网络 (CNN) 路径规划算法, CNN 利用图像信息分析其环境的确切情况, 智能体则通过深度Q学习分析的情况进行导航, 但是文章没有考虑系统拥塞问题; 文献 [10] 提出了一种全局引导的强化学习方法 (G2RL), 利用全局信息来引导智能体各自决策, 实验结果表明, G2RL 具有很好的泛化能力. 2) 完全通信算法, 如 Nash Q-Learning[11]算法, 该算法在智能体决策期间, 每个智能体均实时掌握其他智能体的信息, 因此, 做出的决策有很好的效果, 但是随着智能体数量的增加, 状态空间呈指数式增长, 造成“维度爆炸”, 因此, 很难扩展到数量较多的多智能体任务; 文献 [4] 将 AGV 路径规划问题转化为马尔可夫决策过程, 基于强化学习和 Petri网模型训练神经网络来估计给定多 AGV 系统的奖励函数, 此外, 还提出了一种基于课程学习的技巧加快训练, 但是实验结果表明当 AGV 数量增加至 6 时, 路径规划的成功率大大降低了. 3) CTDE (centralized training and decentralized execution) 算法, 此类算法在训练时, 智能体的所有信息均要上传给中央, 是完全通信的, 而在决策时, 每个智能体均有一个策略网络并根据自身观测选择动作, 不需要通信. 在合作类型任务中, 每个智能体均有共同的目标, 它们的奖励函数是共享的, 在训练时, 每个智能体共同训练一个价值网络. 但是同样是完成了任务, 每个智能体做出的贡献是不一样的, 这就会出现“懒惰智能体”. 为解决这一问题, QMIX[12] (monotonic value function factorisation for deep multi-agent reinforcement learning) 算法提出了一个全局的Q混合网络用于将Q值根据贡献分解给每个智能体; 而在非合作类型任务中, 每个智能体均有自己的目标和奖励函数, 在训练时, 每个智能体均训练自己的价值网络, 学习自己的最优策略, 最终达到纳什均衡. 文献 [13] 对 AGV 冲突情况搭建了整数规划模型, 提出了一种非合作的 MADDPG (multi-agent deep deterministic policy gradient) 方法来解决 AGVs 的无冲突路径规划, 得到了较好的结果.

为了解决多 AGV 调度系统在动态环境中的路径规划问题, 本文将 Petri 网与多智能体强化学习算法相结合, 提出一种避免 AGV 间冲突的 Petri 网建模方法, 在此基础上, 建立一种多智能体非合作深度强化学习路径规划框架. 该框架在避免冲突的前提下实现 AGV 的分布式决策、集中式训练. 在分布式决策中, 本文将路径的分支路口而不是 AGV 视为智能体, 以提升神经网络的泛化能力. 同时, 在设计奖励函数时, 加入 Petri 网环境中 AGV 间的约束条件作为反馈, 从而减少路径规划时的拥塞, 增加规定时间内 AGV 运输货物总量.

本文的主要内容如下: 1) 将 Petri 网建模与强化学习相结合, 提出一种新的建模方法, 解决多 AGV 系统中的冲突避免问题, 增强系统的调度能力; 2) 将路径分支路口视作智能体而非 AGV 本身, 有助于提升神经网络的泛化能力, 允许网络学习到更多关于路径选择的通用模式, 而不是仅局限于特定 AGV 的行为; 3) 将时间成本转化为 Petri 网资源库的限制条件, 并整合至奖励函数, 有效减少系统中的拥塞现象; 4) 在动态变化环境中, 提出一种智能体通过 Nash Q学习来优化自身策略的方法, 以适应环境变化并提高决策质量.

1 问题描述

根据实际货物配送车间搭建拓扑地图, 其可定义为$G=(V,E)$. 其中: $V=\{v_1,v_2,\dots,v_n\}$为地图中的节点 (路口) 集合; E为地图中边的集合, 里面的元素e_{ij}表示一条从节点v_i到节点v_j的有向边. 简易仓库模型如图 1 所示. 图 1 中: 右下角为 AGV 车库, 可供 AGV 充电; 正方形、圆以及三角形均为节点, 每个节点表示 AGV 可以停车和选择动作的地方, 三角形节点为货物的装载点, 正方形节点为货物的卸载点. 货物运输的主要流程如下.

step 1: 有货物到达装载口时, 向随机一辆空载的 AGV 发送运输请求.

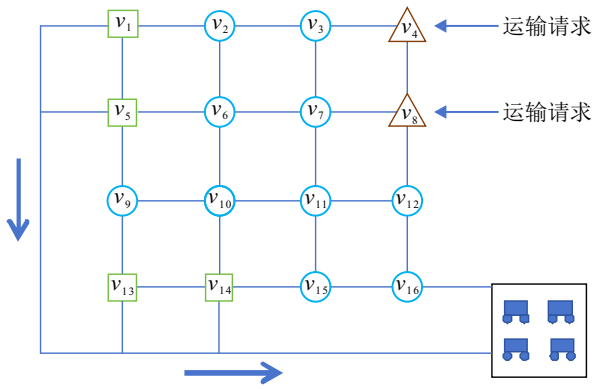


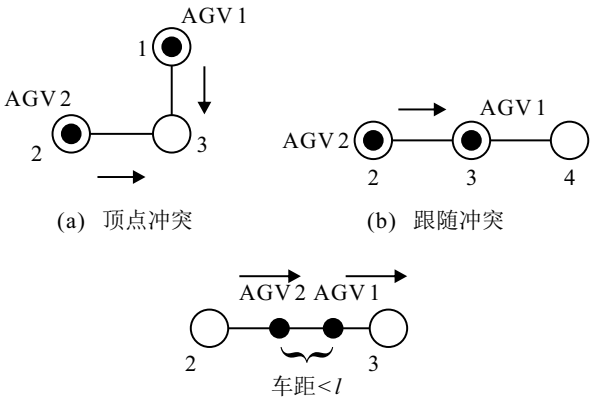
图1 简易物流运输系统

step 2: AGV 接收到请求后, 立即前往对应的装载口装载货物.

step 3: AGV 装载货物完成后, 扫描货物类型, 开始运输货物到指定的卸载口.

step 4: AGV 在卸载口卸载货物后, 通过图 1 中箭头所指的路径返回车库.

上述物流运输系统中会引起以下冲突问题: 1) 节点冲突, 如图 2(a) 所示, AGV1、AGV2 同时前往节点 3; 2) 跟随冲突, 如图 2(b) 所示: AGV2 前往节点 3 与停留在节点 3 的 AGV1 碰撞; 3) 边冲突, 如图 2(c) 所示: 一条边上出现了 1 辆以上的 AGV, 意味着 AGV 间的距离小于安全距离l.

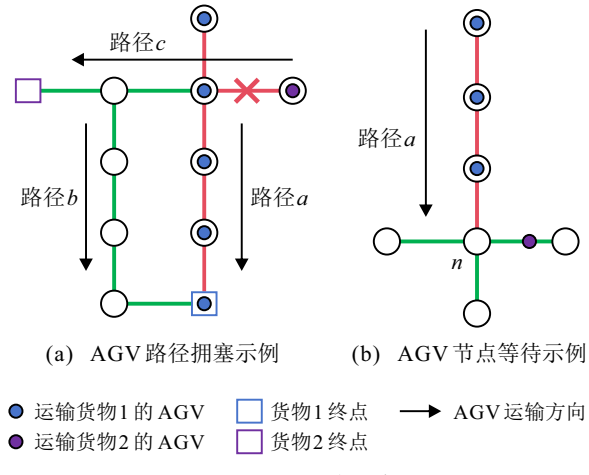


(a) 顶点冲突 (b) 跟随冲突

(c) 边冲突

图2 AGV 冲突问题

针对上述冲突, 可在路径中设置 AGV 可能的停留点, 确保 AGV 间的距离大于等于l, 如图 3 所示. 其中: 圆圈为可能的停留点或路径分支点, 正方形为目的地, 两个圆圈间或圆圈与正方形间的道路称为边. 停留点、分支点以及目的地只能停靠 1 辆 AGV, 每条边上只能行驶 1 辆 AGV. 然而, 随着 AGV 数量的增加, 不恰当的路径规划会导致拥塞问题, 如图 3(a) 所示: 运输货物 1 的 AGV 均通过路径a运输货物, 造成路径a拥塞, 这不仅降低了路径a中 AGV 的运



(a) AGV 路径拥塞示例 (b) AGV 节点等待示例

图3 多 AGV 系统拥塞问题

输效率, 还阻塞了路径c中运输货物 2 的 AGV, 增加了总运输时间. 另外, 如图 3(b) 所示: 当 1 辆 AGV 前往节点n (路口) 时, 会导致路径a中的其他 AGV 需要等待该 AGV 离开节点n.

本文主要工作内容如下: 1) 解决 AGV 间的冲突问题; 2) 规划 AGV 到达终点的最优路径, 减少多 AGV 系统中的拥塞.

2　AGV 系统无冲突模型搭建

2.1　Petri 网

Petri 网N可由一个五元组$N=(P,T,W,F,M)$表示, 其结构如图 4 所示. 其中: $P=\{p_1,p_2,\dots,p_n\}$为一组有限的库所集合; $T=\{t_1,t_2,\dots,t_m\}$为一组有限的变迁集合; $F\subseteq(P\times T)\bigcup(T\times P)$为库所与变迁相连的有向弧集合; W为 Petri 网N的权重函数, 为每个弧指定一个正整数权重, 可表示为$W:F\to Z^+$, 本文设置弧的权重函数为 1; M为 Petri 网N的一个标识, 是一个$1\times|P|$的向量, 用于表示各库所中令牌数量的分布, $M(p_i)$为库所p_i中的令牌个数, 若库所p_i为空, 则称其有一个空的标记, 用$M(p_i)=0$表示, 每个库所中均有非负的令牌数量, 即$M(p_i)>0$.

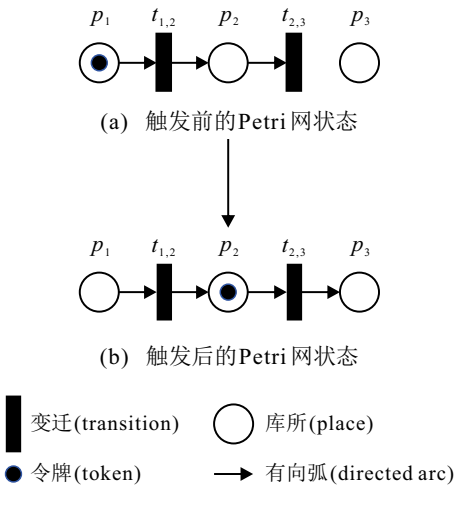


图4　Petri 网结构以及状态转换

前关联矩阵A^-为$|T|\times|P|$的矩阵, 对于所有$(p,t)\in P\times T$, 若$(p,t)\subseteq F$, 则$A^-(t,p)=1$; 否则, $A^-(t,p)=0$. 后关联矩阵A^+为$|T|\times|P|$的矩阵, 对于所有$(t,p)\in T\times P$, 若$(t,p)\subseteq F$, 则$A^+(t,p)=1$; 否则, $A^+(t,p)=0$. 关联矩阵$A=A^+-A^-$, 用于表示变迁触发引起的库所间的令牌转换. 若一个变迁t要触发, 则需要满足以下条件:

$$M_k(p_i)-A^-(t,p_i)\geqslant 0, \tag{1}$$

其中$M_k(p_i)$表示k时刻库所p_i持有的令牌数量. 以图 4 为例, 图 4(a) 中的$t_{1,2}$想要触发, 则需要满足与$t_{1,2}$相连的前置库所中至少有一个令牌, 即$M(p_1)\geqslant 1$. 当图 4(a) 中变迁$t_{1,2}$触发时, Petri 网的状态由图 4(a) 转化为图 4(b), 即p_1的令牌通过$t_{1,2}$转移到了p_2. 具体的令牌转移公式如下所示:

$$M_{k+1}(p_i)=M_k(p_i)+A(t,p_i), \tag{2}$$

其表示k时刻触发了t变迁, 库所p_i的令牌分布由$M_k(p_i)$变为$M_{k+1}(p_i)$.

2.2　基础模型

Petri 网对实际路段进行建模, 实际路段如图 5 所示. 图 5(a) 中: 箭头表示有向路径, 长方形表示路径分支点, 路径分支点连接多条不同方向的路径 (未画出). 设置 AGV 间的最短间隔距离为l m. 根据最短间隔距离在路径上设置 AGV 停留点, 假设路径长度为s m, 则应设置停留点数目为$(s-l)/l$个. 图 5(a) 路径设置停留点如图 5(b) 所示.

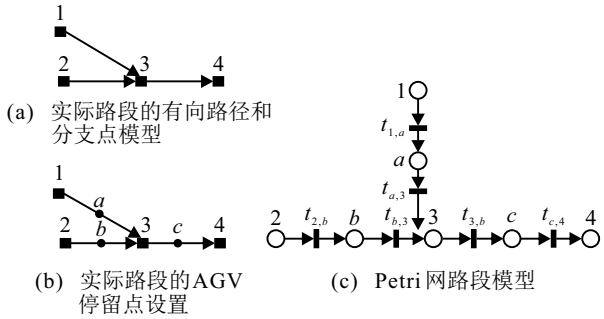


图5　实际路段

Petri 网搭建如图 5(c) 所示的路段模型, Petri 网库所对应实际路段中的路径分支点和 AGV 停留点, 库所中的令牌数量对应 AGV 的数量. 每个路径分支点和 AGV 停留点允许至多 1 辆 AGV 停留, 即$M(p_i)\leqslant 1$. AGV 行驶在边上, 未到达节点 (路径分支点或停留点), 仍然视为在上一节点.

2.3　无冲突模型

考虑到 AGV 间的安全性问题, 本文在原有的 Petri 网基础上, 加入了监控库所, 组成 Petri 网$G=(N,P_R)$. 其中: N为 Petri 网, 用于搭建实际路段的映射模型; P_R为有限的监控库所集合, 用于限制 AGV 的行动, 避免冲突产生.

本文中每个普通库所均有 1 个监控库所, Petri 网中每个监控库所的初始令牌数为 1, 即$M_0(P_{R_i})=1$.

加入监控库所的 Petri 网如图 6 所示. 此时, 图 6(a) 中的 AGV1 和 AGV2 同时前往点 3, 需要抢夺点 3 对应监控库所的令牌. 优先触发变迁获得监控库所中令牌的 AGV 可以执行其动作, 而另一 AGV 则需要尝试获取点 3 对应监控库所的令牌或选择其他动作触发其他变迁.

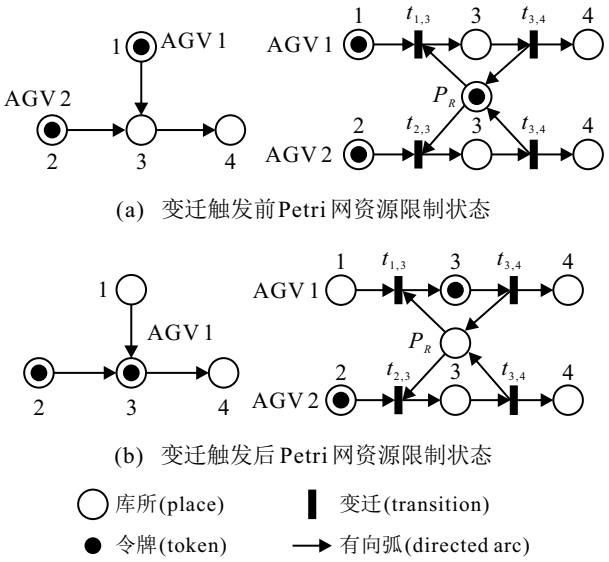


图6 AGV 运动路线以及加入监控库所的 Petri 网

监控库所的前关联矩阵和后关联矩阵B^-、B^+均为$|T|\times|P_R|$的矩阵. 关联矩阵为$B=B^+-B^-$, 其具体定义如下所示:

$$B^+(t_j,p_{R\_i})=0,$$
$$B^-(t_j,p_{R\_i})=w(p_{R\_i},t_j)-w(t_j,p_{R\_i}). \tag{3}$$

加入监控库所后的 Petri 网, 其变迁触发条件不仅要遵循式 (1), 还要受到监控库所的制约, 限制条件如下所示:

$$M_k(P_{R\_i})-B^-(t,P_{R\_i})\geqslant 0. \tag{4}$$

其中: $M_k(P_{R_i})$为k时刻库所P_{R_i}中令牌的数量, 监控库所P_{R_i}的状态更新公式为

$$M_{k+1}(P_{R\_i})=M_k(P_{R\_i})+B(t,P_{R\_i}). \tag{5}$$

3 Petri 网环境中基于 MADDPG 算法的 AGV 最优方向控制器

3.1 路口控制器设计

为了减少拥塞, 实现 AGV 的最优路径规划, 将 Petri 网中 AGV 的路径规划问题视为部分可观测的马尔可夫决策过程 (POMDP), 如图 7 所示: 路口分支点作为智能体通过自身观测o_i为 AGV 提供动作a_i, 将动作a_i映射为 Petri 网中的变迁t_i, Petri 网的监控库所判断t_i能否执行并给予奖励r_i, 执行动作后 AGV 到达下一路口并获得新的观测. 智能体将不断重复上述过程, 直至训练出最优策略.

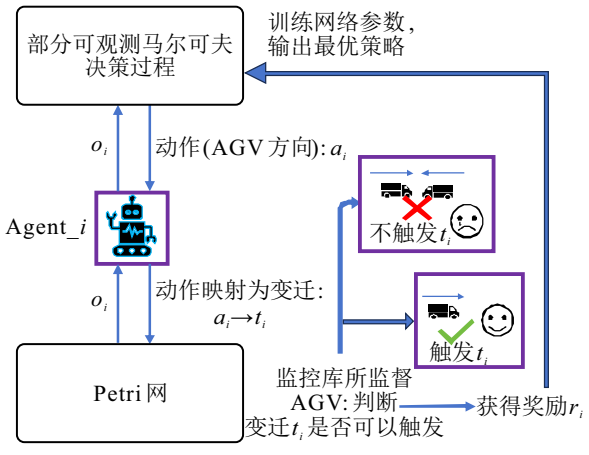


图7 Petri 网环境的 POMDP

定义局部观测空间: 路口分支点通过策略网络为停留的 AGV 提供方向, 策略网络的输入为路口分支点的局部观测空间, 其包括当前路口分支点信息、相连可达路口分支点信息以及可达路径上停留点信息. 如图 8 所示: 路径分支点 3 的观测空间$o_3=[s_1,s_2,s_3,s_4,s_a,s_b,s_c]$. 其中: $\{a,b,c\}$为路径中的停留点, $\{1,2,3,4\}$为路口分支点. 下式为智能体i局部观测空间的定义:

$$s_i=\begin{cases}(\alpha_1,\alpha_2,\ldots,\alpha_m,\ldots,\alpha_n),\ i\in N_t;\\ \beta,\ i\in N_p.\end{cases}$$

$$\begin{cases}\alpha_x=0,\ x=[1,2,\ldots,n],\ \text{不存在AGV};\\ \alpha_x=\begin{cases}1,\ x=m;\\ 0,\ x\neq m;\end{cases}\ \text{存在AGV, 其任务为}m.\end{cases}$$

$$\begin{cases}\beta=0,\ \text{不存在AGV};\\ \beta=1,\ \text{存在AGV}.\end{cases} \tag{6}$$

其中: N_t为路口分支点的集合, N_p为路径中停留点的集合, s_i为路口分支点i或路径停留点i的信息.

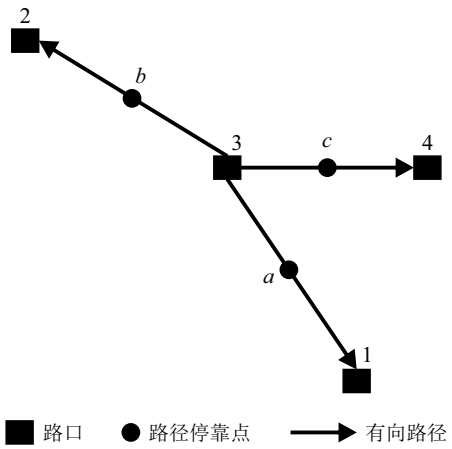


图8 路径分支点的部分可观测定义

定义动作空间: 路径分支点为 AGV 提供方向以到达下一个路口分支点, 其动作空间的维度与可达的路口数量有关, 若可达的路口数量为n, 则其动作空间的维度为n.

定义奖励函数: 智能体i的奖励函数R_i主要由 3 部分组成, 如下所示:

$$R_i=k_1\cdot r_1^+ + k_2\cdot r_1^- + k_3\cdot r_2 + k_4\cdot r_3.$$

$$r_1=\begin{cases}1,\ \text{满足触发条件(4)};\\ -1,\ \text{不满足触发条件(4)};\end{cases}$$

$$r_2 = \varphi(i,d) - \varphi(j,d);$$
$$r_3 = \begin{cases} 10, & j = d; \\ 0, & j \neq d. \end{cases} \tag{7}$$

其中: r_1为一个惩罚项, 表示智能体为 AGV 提供的动作会与其他 AGV 产生冲突的惩罚, 即判断采取的行动是否满足式 (4), 若不满足, 则$r_1 = -1$, 记为r_1^-; 否则$r_1 = 1$, 记为r_1^+. 该惩罚项的设置不仅避免了 AGV 碰撞产生, 还可令节点在当前状态下采取的动作尽可能避免拥堵的产生. r_2用于比较当前路口和上一路口分别与终点的最短可达距离. r_3为动作触发后到达终点所给予的奖励. 当 AGV 在十字路口选择动作, 但是不满足式 (4) 时, 该动作将不执行, 此时, 权重系数$k_2 = 1$, k_1、k_3、$k_4 = 0$; 否则, 权重系数k_3、$k_4 = 1$, k_1和k_2分别为过程中从当前路口到下一个路口动作触发和未触发的次数. $\phi(i,d)$为当前路口i到终点d的最短可达距离. $\phi(j,d)$为下一路口j到终点d的最短可达距离.

3.2　非平稳环境下的纳什Q更新

本文采用多智能体强化学习方法来求解路口控制器的纳什均衡策略[14]. 在纳什均衡下, 每个智能体在已知其他智能体策略的情况下选择最有利于自己的应对策略, 即在联合状态$S(t)$下遵循下式:

$$V_i(S(t),\pi_1^*,\ldots,\pi_n^*) \geqslant V_i(S(t),\pi_1^*,\ldots,\pi_i,\ldots,\pi_n^*), \quad \forall \pi_i \in A_i. \tag{8}$$

其中: $V_i(S(t),\pi_1^*,\ldots,\pi_n^*)$为智能体$i$在纳什均衡策略下的回报, π_i^*为智能体i的最优策略, π_i为智能体i当前策略, A_i为智能体i的策略集合.

在非平稳环境中, 由于智能体采用分布式决策而不是集中式决策, 智能体的策略并不是同步产生的, 且智能体的奖励函数只能通过自身动作来获得, 因此, 将 Nash Q函数定义为

$$Q_i^*(s_t,a_t^1,\ldots,a_t^n) = r_i(s_t,a_t^i) + \gamma \cdot \sum p(s_{t+1}|s_t,a_t^1,\ldots,a_t^n) \cdot v_i(s_{t+1},\pi_1,\ldots,\pi_i^*,\ldots,\pi_n). \tag{9}$$

其中: γ为折扣因子; $p(s_{t+1}|s_t,a_t^1,\ldots,a_t^n)$为状态$s_t$转移至状态$s_{t+1}$的概率; $r_i(s_t,a_t^i)$为智能体i在状态s_t下采取动作a_t^i获得的奖励; $v_i(s_{t+1},\pi_1,\ldots,\pi_i^*,\ldots,\pi_n)$为在其他智能体策略不变时, 智能体$i$采取最优策略下的目标回报.

3.3　多 AGV 路径规划

MADDPG 算法是一种多智能体强化学习算法, 它通过深度学习来适应复杂的环境, 提高路径规划的鲁棒性和适应性. 本文框架采用去中心化执行, 集中式训练的方法, 即每个智能体均有 4 个网络: 策略网络、目标策略网络、价值网络、目标价值网络, 分别用$\mu(o_i;\theta_i)$、$\mu(o_i;\theta_{i-})$、$Q(s;a;\omega_i)$、$Q(s;a;\omega_{i-})$表示. 策略网络是确定性的: 策略网络的输入为智能体的观测值o_i, 对于确定的输入, 输出的动作$a_i = \mu(o_i;\theta_i)$为确定的. 价值网络的输入为全局状态s以及所有智能体的动作a, 输出为一个实数, 表示基于状态s执行动作a的好坏程度. 其中: $s = [s_1, s_2, \ldots, s_n]$, 包含所有节点 (路口分支点和停留点) 的信息; $a = [a_1, a_2, \ldots, a_n]$, 包含所有智能体的动作. 本文采用非合作的 Nash Q学习方法来训练智能体的价值网络, 如图 9 所示. 每个智能体的价值网络根据自己的

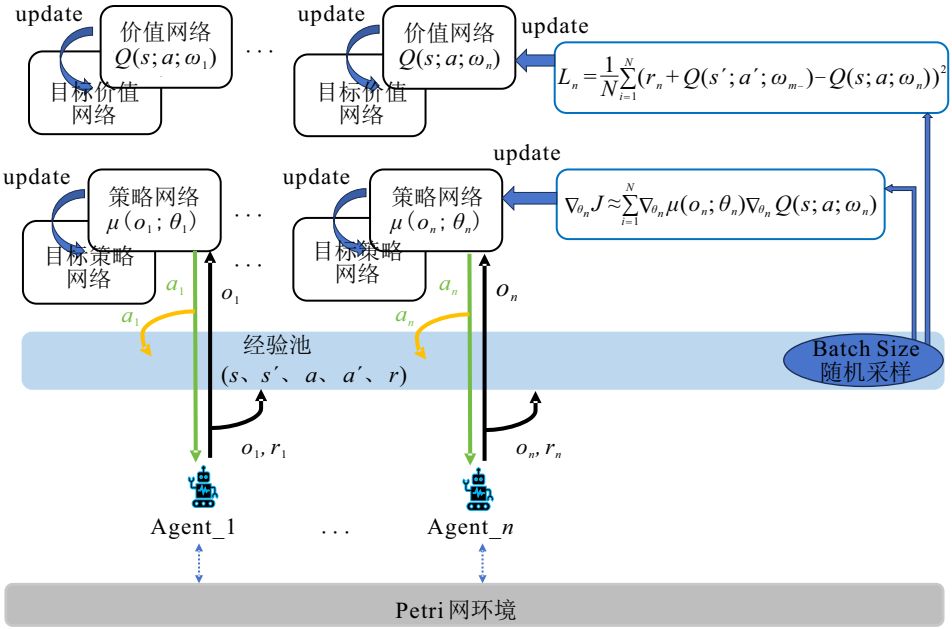


图9　智能体框架

奖励函数来训练, 智能体的价值网络指导自己的策略网络做出改进.

由于传统的 MADDPG 算法要求动作空间是连续的, 而本文的动作空间是离散的. 对此, 使用 Gumbel-Softmax 策略[15] 来获得离散分布的近似采样, 如下所示:

$$y_i = \frac{\exp(\log a_j + g_i)/T}{\sum_{j=1}^{k}\exp((\log a_j + g_i)/T)},\ i = 1, 2, \ldots, k;$$

$$g_i = -\log(-\log u),\ u\ \text{Uniform}(0, 1). \tag{10}$$

其中: g_i为一个采样自 Gumbel (0, 1) 的噪声; T为温度参数, 用于控制生成的分布与离散分布的近似程度, T越大, 生成的分布越趋近于均匀分布, 本文将T设置为 1.

智能体信息采集如图 10 所示. 当 AGV 进入路口n时, 节点 (路口) 智能体会根据观测情况选择 AGV 的行进方向. 同时, AGV 会记录下所有其他智能体在当前时刻的观测和动作, 作为全局观测o、动作a以及状态s. 当该 AGV 到达下一个路口m时, 会利用下一个路口的目标策略网络来预测下一个步骤的动作a'_m, 并将该步骤所有智能体的观测和动作记录为o'和a', 状态为s'. 将上述信息存入智能体n的经验池D_n并开始训练, 如图 9 所示: 通过损失函数L_n来训练路口智能体n的价值网络, 根据目标函数J来更新智能体n的策略网络. MADDPG 算法求解路径规划如算法 1 所示.

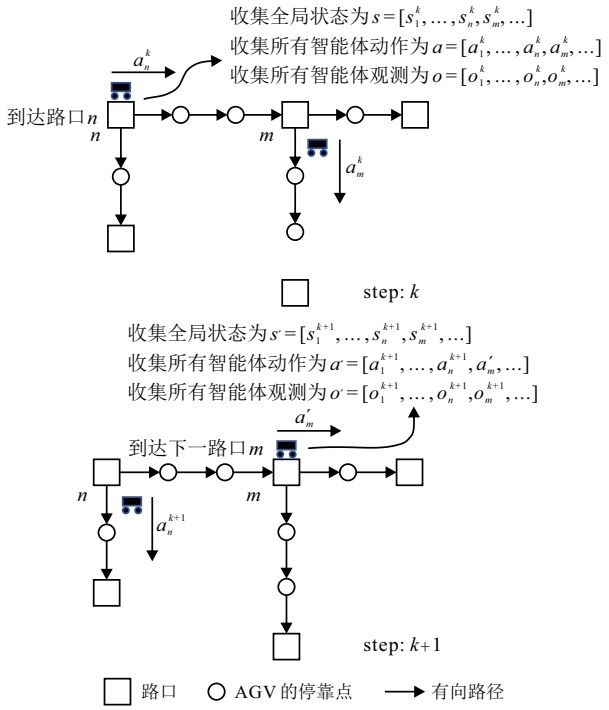


图10 智能体信息收集

算法1 MADDPG算法求解路径规划.

输入: 初始化所有智能体的Actor网络$\mu(o_i;\theta_i)$和Critic网络$Q(s;a;\omega_i)$的参数为θ_i和ω_i, 初始化其目标网络参数$\theta_{i_}$和$\omega_{i_}$为$\theta_{i_}\leftarrow\theta_i$, $\omega_{i\_}\leftarrow\omega_i$; 初始化经验池$D_i$大小为$N$, 设置batch_size; 初始化Petri网监控库所令牌分布$M_0(P_{R_i})=1$.

1. for episode = 1, M do
2. 重置Petri网环境;
3. 获取每个智能体的初始状态;
4. for $t=1$ to T do
5. for 智能体i do
6. AGV进入路口获得动作a_i将其映射为t_i, 判断动作能否执行并给予奖励r_i, 更新Petri网, 同时, AGV收集全局状态s、全局动作a、所有观测o;
7. AGV到达下一路口并获得下一时刻的全局状态s'、自身的观测o'_i, 根据自身观测做出预测动作a'_i, AGV收集全局状态s'、全局动作a'、所有观测o';
8. end for
9. 存储经验(s、a、a'、r、s')至经验池D_i, 获得经验池总量为n;
10. if $n\geqslant$ batch_size then
11. 每个智能体从经验池D_i中随机采样batch_size大小的(s、a、a'、r、s')数据;
12. 根据图9损失函数L_i更新价值网络参数ω_i;
13. 根据图9目标函数J更新所有策略网络参数θ_i;
14. 更新目标网络参数:

$$\theta_{i\_}\leftarrow\tau\theta_i+(1-\tau)\theta_{i\_}$$

$$\omega_{i\_}\leftarrow\tau\omega_i+(1-\tau)\omega_{i\_}$$

15. end if
16. end for
17. end for

输出: 训练后的策略网络参数θ_i^*.

参数设置: 经验池D_i容量$N=100$, batch_size大小为 16, 每个策略网络和价值网络的学习率均为 0.000 5, 折扣因子$\gamma=0.99$, $\tau=0.005$.

4 仿真实验与分析

针对基于 Petri 网和深度强化学习多 AGV 路径规划进行仿真, 采用 Python 3.9、Pytorch 和 AnyLogic 软件对所提出框架算法进行仿真, 验证多 AGV 路径规划算法的有效性.

实验对 AGV 在物流运输系统的路径规划问题, 制定如下规则.

1) 允许至多 1 辆装载货物的 AGV 停在路口或停靠点, AGV 间的距离大于等于10 m;

2) 所有 AGV 的型号相同, AGV 的最大行驶速

度为 2 m/s, 装载和卸载货物时间为 8 s;

3) 卸载口随机生成不同类型的货物, 货物生成后, 向空载 AGV 发送运输请求, 请求不能被抢占.

图 11 为 AnyLogic 软件搭建的物流运输系统. 初始化时, AGV 均停靠在车库中. 货物从进货口产生向空载的 AGV 发送运输请求, AGV 接收任务请求后前往装载口装载货物并运输至指定地点 (共 8 个目的地, 对应图 11 中 4 个出货口和 4 个仓库). 运输完成后, AGV 从 pth 路径返回至车库等待下一次任务请求.

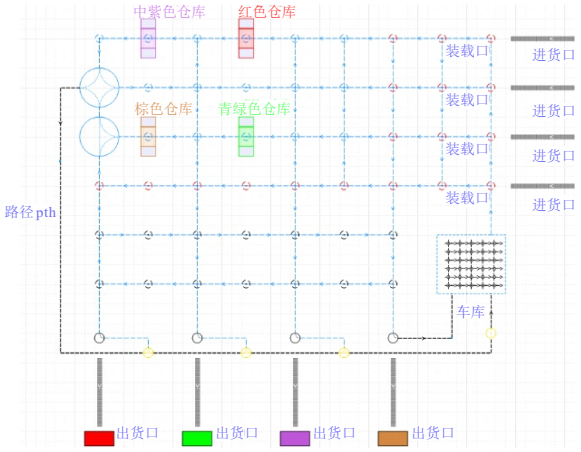


图11　AnyLogic 物流运输系统

图 12 为 MADDPG 算法下, AGV 运输货物的实际情况. 仿真环境为动态环境, 每一时刻 Petri 网环境中的 AGV 数量是动态变化的, 进货口生成的货物种类也是随机的, 但是, 任务目的地是固定不变的.

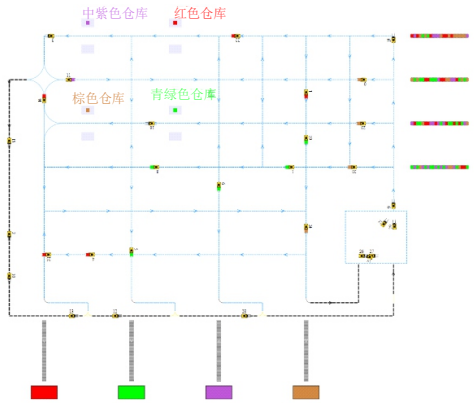


图12　AGV 实际路况

首先, 上述系统中的任务是随机变化的, 且 AGV 的数量也是动态变化的, 这会导致系统的总目标不断变化, 因此, 无法简单地使用一个固定的总奖励函数来衡量系统的收敛性. 图 13 为系统总损失值. 尽管存在动态变化, 通过观察图 13 中所有节点智能体的总损失值下降情况, 可以发现系统在优化过程中的进展. 由于在奖励函数中引入了曼哈顿距离, 这一因素促使每个智能体能够快速收敛至较优的决策路径. 然而, 由于任务分配和 AGV 数量的随机性, 节点智能体必须持续调整其策略来适应这些变化, 以确保系统整体性能的最优化.

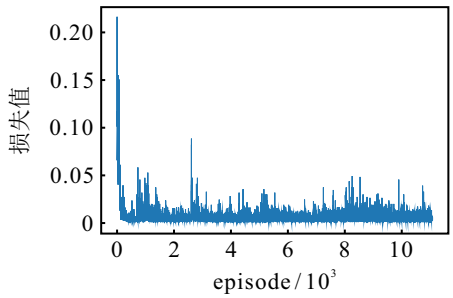


图13　系统总损失值

图 14 为两种不同模型下的数据. 如图 14 所示: 蓝色数据模型在训练时, 其奖励函数中加入了 Petri 网触发时的反馈, 即式 (10) 中r_1; 红色数据模型在训练时, 其奖励函数中没有加入 Petri 网触发时的反馈. 图 14 中分别对比了不同数量 AGV 在规定时间 (28 800 s) 下运输货物的总量. 由图 14 可见: 当 AGV 数量为 10 时, 两种模型的送货总量基本相近, 随着 AGV 数量的增加, 两种模型运输货物总量的差距明显增加, 表明 AGV 数量的增加会加大系统出现拥塞的可能性. 同时, 实验也验证了在奖励函数中加入 Petri 网触发时的反馈可有效地减少运输货物时拥塞的发生.

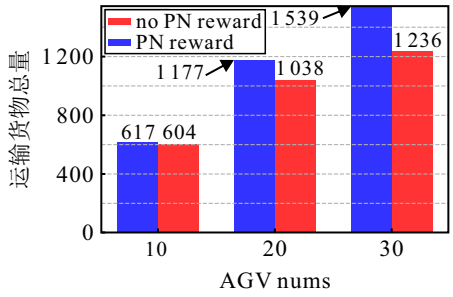


图14　Petri 网奖励对 AGV 运输货物总量影响

图 15 为 MADDPG 算法与 QMIX 算法在规定时间 (28 800 s) 内的运输货物总量对比. 由图 15 可见, 随着 AGV 数量增多, MADDPG 模型下的送货效

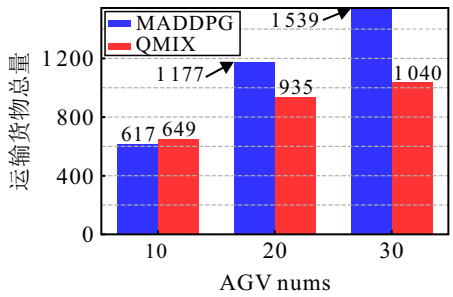


图15　不同算法对 AGV 运输货物总量影响

率远高于 QMIX 算法的模型. 此外, 通过实验发现: QMIX 算法易陷入局部最优, 训练结果也易出现 “绕自行车” [16] 的现象, 增加了系统死锁的风险.

5 结 论

本文解决了多 AGV 的动态路径规划的问题, 提出了一种 Petri 网建模的方法, 有效地防止了 AGV 间的冲突产生, 在此基础上, 搭建了基于深度强化学习的多 AGV 路径规划框架, 将路径规划问题转化为部分可观测的马尔可夫决策过程, 以路径分支点为智能体设计了动作空间、状态空间、观测空间和奖励函数, 在奖励函数中加入了 Petri 网的触发反馈, 极大程度地减少了 AGV 运输货物时拥塞的产生, 增加了系统模型在规定时间内的送货总量. 最后, 在 AnyLogic 软件中验证了所提出算法的有效性.

参考文献 (References)

[1] Nishi T, Maeno R. Petri net decomposition approach to optimization of route planning problems for AGV systems[J]. IEEE Transactions on Automation Science and Engineering, 2010, 7(3): 523-537.

[2] Eda S, Nishi T, Mariyama T, et al. Petri net decomposition approach for Bi-objective routing for AGV systems minimizing total traveling time and equalizing delivery time[J]. Journal of Advanced Mechanical Design, Systems, and Manufacturing, 2012, 6(5): 672-686.

[3] Luo J L, Wan Y X, Wu W M, et al. Optimal Petri-net controller for avoiding collisions in a class of automated guided vehicle systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(11): 4526-4537.

[4] Zhang H B, Luo J L, Lin X J, et al. Dispatching and path planning of automated guided vehicles based on Petri nets and deep reinforcement learning[C]. IEEE International Conference on Networking, Sensing and Control. Xiamen, 2021: 1-6.

[5] Smolic-Rocak N, Bogdan S, Kovacic Z, et al. Time windows based dynamic routing in multi-AGV systems[J]. IEEE Transactions on Automation Science and Engineering, 2010, 7(1): 151-155.

[6] Connell D, La H M. Dynamic path planning and replanning for mobile robots using RRT[C]. IEEE International Conference on Systems, Man, and Cybernetics. Banff, 2017: 1429-1434.

[7] Tan M. Multi-agent reinforcement learning: Independent vs. cooperative agents[C]. Proceedings of the 10th International Conference on Machine Learning. Amherst, 1993: 330-337.

[8] Hwang I, Jang Y J. $Q(\lambda)$ learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs[J]. International Journal of Production Research, 2020, 58(4): 1199-1221.

[9] Bae H, Kim G, Kim J, et al. Multi-robot path planning method using reinforcement learning[J]. Applied Sciences, 2019, 9(15): 3057.

[10] Wang B Y, Liu Z, Li Q B, et al. Mobile robot path planning in dynamic environments through globally guided reinforcement learning[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6932-6939.

[11] Hu J, Wellman M P. Nash Q-learning for general-sum stochastic games[J]. Journal of Machine Learning Research, 2004, 4(6): 1039-1069.

[12] Tabish R, Mikayel S, de Schroeder W C, et al. Monotonic value function factorisation for deep multi-agent reinforcement learning[J]. Journal of Machine Learning Research, 2020, 21(1): 7234-7284.

[13] Hu H T, Yang X R, Xiao S C, et al. Anti-conflict AGV path planning in automated container terminals based on multi-agent reinforcement learning[J]. International Journal of Production Research, 2023, 61(1): 65-80.

[14] Khan M A, Sun Y N. Chapter 46 non-cooperative games with many players[J]. Handbook of Game Theory with Economic Applications, 2002, 3: 1761-1808.

[15] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J/OL]. 2017, arXiv: 1706.02275.

[16] Randløv J, Alstrøm P. Learning to drive a bicycle using reinforcement learning and shaping[C]. Proceedings of the 15th International Conference on Machine Learning. Madison, 1998: 463-471.

作者简介

于绍琪 (1999–), 男, 硕士生, 主要研究方向为 Petri 网结合强化学习的路径规划, E-mail: 935804494@qq.com;

田玉平 (1964–), 男, 教授, 博士, 博士生导师, 主要研究方向为复杂系统控制的理论、方法与工程应用, 通信网络中的优化与控制、混沌控制及其在通信及信息处理中的应用、智能机器人与控制, E-mail: tianyp@hzcu.edu.cn.