

基于联合 Q 值分解的强化学习网约车订单派送

黄晓辉, 张 雄, 杨凯铭, 熊李艳

(华东交通大学 信息工程学院, 南昌 330013)

摘 要: 因网约车订单派送不合理, 导致资源利用率和出行效率降低。基于联合 Q 值函数分解的框架, 提出两种订单派送方法 ODDRL 和 LF-ODDRL, 高效地将用户订单请求派送给合适的网约车司机, 尽可能缩短乘客等待时间。为捕获网约车订单派送场景中随机需求与供应动态变化关系, 把城市定义为一张四边形网格的地图, 将每辆车视为一个独立的智能体, 构建多智能体马尔可夫决策过程模型, 通过最大化熵与累计奖励训练智能体。将多智能体的联合 Q 值函数转化为易分解函数, 使联合 Q 值函数与单个智能体值函数中的动作具有一致性, 同时设计动作搜索函数, 结合集中训练、分散执行策略的优点, 让每辆车以分布式的方式解决订单匹配问题, 而不需要与其他车辆进行协调, 从而降低复杂性。实验结果表明, 相比 Random、Greedy、QMIX 等方法, 所提 ODDRL 和 LF-ODDRL 具有较优的扩展性, 其中, 在 500×500 网格上, 当乘客数为 10、车辆数为 2 时, 相对于 QMIX 方法接送乘客所产生的总时间分别缩短 5% 和 12%。

关键词: 多智能体; 强化学习; 值函数; 订单派送; 神经网络

开放科学(资源服务)标志码(OSID):



中文引用格式: 黄晓辉, 张雄, 杨凯铭, 等. 基于联合 Q 值分解的强化学习网约车订单派送[J]. 计算机工程, 2022, 48(12): 296-303, 311.

英文引用格式: HUANG X H, ZHANG X, YANG K M, et al. Reinforcement learning online car-hailing order dispatch based on joint Q -value decomposition[J]. Computer Engineering, 2022, 48(12): 296-303, 311.

Reinforcement Learning Online Car-Hailing Order Dispatch Based on Joint Q -value Decomposition

HUANG Xiaohui, ZHANG Xiong, YANG Kaiming, XIONG Liyan

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

[Abstract] Resource utilization and travel efficiency are often reduced owing to an unreasonable dispatch of online car-hailing orders. Based on the joint Q -value function decomposition framework, two order dispatch methods, ODDRL and LF-ODDRL, are proposed to efficiently dispatch user requests to appropriate online car-hailing drivers to minimize passenger waiting times. To capture the dynamic change relationship between random demand and supply in the online car-hailing order dispatch scenario, the city is defined as a quadrilateral grid map, and each vehicle is considered as an independent agent. A multi-agent Markov Decision Process (MDP) model is developed to train agents by optimizing entropy and cumulative rewards. The joint Q -value function of multi-agents is transformed into a decomposable function so that the actions in the joint Q -value function and the value function of a single agent are consistent. At the same time, the action search function is designed by combining the benefits of centralized training and decentralized execution strategy so that each vehicle can solve the order matching problem in a distributed manner without coordinating with other vehicles, thereby reducing complexity. The experimental results demonstrate that the proposed ODDRL and LF-ODDRL have better scalability than Random, Greedy, QMIX, and other methods. On the 500×500 grid, when the number of passengers is 10 and the number of vehicles is 2, the total time for picking up is shortened by 5% and 12% respectively, when compared to the QMIX method.

[Key words] multi-agent; reinforcement learning; value function; order dispatch; neural network

DOI: 10.19678/j.issn.1000-3428.0063438

基金项目: 国家自然科学基金(62062033, 62067002, 61967006); 江西省自然科学基金青年重点项目(20192ACBL21006); 江西省自然科学基金面上项目(20212BAB202008)。

作者简介: 黄晓辉(1984—), 男, 副教授、博士, 主研方向为深度学习、智慧交通; 张 雄、杨凯铭, 硕士研究生; 熊李艳, 教授。

收稿日期: 2021-12-02 **修回日期:** 2022-01-13 **E-mail:** 1056251609@qq.com

0 概述

随着移动互联网技术的发展以及智能手机的普及,依托移动互联网平台的共享经济在各个行业逐渐涌现,并对人们的生活方式产生较大的影响。在交通出行领域,研究人员提出基于网约车平台的多种新型出行方式。文献[1]介绍网约车平台通过整合乘客与网约车之间的供需信息使其相匹配。网约车平台可以提高当前的交通效率,当乘客用智能手机设置目的地时,平台可自动采用智能方法匹配可用车辆。由于订单派送结果直接影响平台收入及乘客舒适度,因此订单派送问题也可以类比为PDP(Pickup and Delivery Problem)问题,其关键是将订单与可用车辆进行合理匹配。因此,通过合适的匹配方法来派送订单,使车辆的总行驶距离和用户的等待时间最小化,并增加司机的收入。

高效的订单派送和空间扩展性是构建智能派单系统的关键。如果一辆汽车绕道去接送客户,并且在接送过程中,司机还需考虑堵车问题,用户的总出行时间就会延长。文献[2]根据出行路径,将网约车与订单进行匹配,利用强化学习对司机的出行路径进行决策,同时,通过训练历史出行路径,提高出行效率,增加司机收益。但是该方法存在一定的局限性,在训练历史出行路径数据时仅考虑网约车到乘客初始位置的状态,因乘客出行会改变驾驶员的位置,导致网约车司机的未来收益减少。对于空间扩展性,文献[3]提出基于深度强化学习的网约车重定位原理,提前将网约车进行需求调度,考虑到一次性优化整个网约车平台的复杂性以及不同规模城市之间供需情况的差异,通常会划分自然地理边界来解决订单派送问题。由于每个城市具有不同的规模和交通模式,因此在网约车平台上为每个城市构建新的模型。

本文结合深度神经网络与多智能体强化学习(MARL),在基于联合 Q 值函数分解的框架VFD下,提出两种方法ODDRL与LF-ODDRL,使乘客整体等待时间最小化。通过将订单派送建模为分布式决策问题,具有集中训练、分散执行策略的优点,同时设计基于动作搜索的损失函数,将订单高效地与网约车司机相匹配,提高模型在不同规模网格上的可扩展性。

1 相关工作

1.1 基于组合优化的PDP方法

网约车订单派送体系结构如图1所示。车辆与订单的合理匹配是网约车平台中的一项重要决策任务。文献[4]考虑为每个订单找到最近司机。文献[5]通过网约车平台给司机发送虚拟订单,使网约车提前调度到未来乘客可能多的地方,满足更多乘客的需求,提高司机收益。文献[6]使用人工编写的

函数来估计出行时间和用户等待时间。文献[7]提出新型双特征协同转换器,通过新的循环位置编码方法嵌入车辆位置特征,使用Transform设计有效的车辆路径解决方案。文献[8]提出基于自注意力机制的深层架构,并将其作为策略网络来引导下一个动作的选择,解决旅行商路径最短问题。文献[9]提出基于集中组合优化的新模型,该模型在短时间内将订单并发匹配多个车辆。但是上述方法大多需要计算所有可用的车辆派送订单,并对每个城市构建新的模型,难以用于大规模的网约车订单派送,扩展性较差。

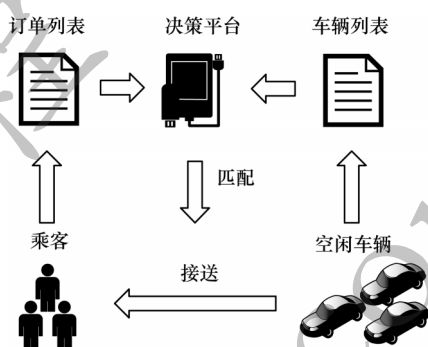


图1 网约车订单派送体系结构

Fig.1 Structure of online car-hailing order dispatch system

1.2 基于强化学习的PDP方法

文献[10]描述了PDP问题如何从一个组合优化方法发展到一个包含半马尔可夫决策过程(Markov Decision Process, MDP)模型的深度强化学习方法。文献[11]将神经网络与异构注意力机制相结合,增强深度强化学习中的策略,侧重关注乘客的上车位置,该策略期望寻找网约车订单派送问题中的最优解。文献[12]利用model-free强化学习来解决最优解问题,通过与复杂环境交互来学习策略。文献[13]将某一个区域的司机视为具有相同状态的智能体,简化了大规模订单派送场景中的随机供需动态。文献[14]考虑对异构车队约束的车辆选择解码器和基于路由结构的节点选择解码器,通过在每一步自动选择车辆和节点来构建解决策略,使得车辆的总行驶时间最小化。文献[15]通过深度强化学习理论对智能体的路径规划策略进行改进,文献[16]基于知识迁移学习方法,通过深度神经网络估计驾驶员的状态动作值函数,利用训练历史数据来达到跨多个城市的模型迁移目的。文献[17]提出一种新的方法,将多智能体强化学习转化为两个智能体不断交互的过程,使其在大规模智能体中更好地学习策略。文献[18]基于独立DQN的方法学习分散的价值函数。文献[19]基于深度价值网络的批量训练方法来学习时空状态值函数,通过基于价值的策略搜索,将规划和引导与价值网络相结合,按需生成最优的重新定位行动。

在上述方法的基础上,本文基于VFD框架来解决订单派送问题,VFD框架由联合动作价值网络、独立动作价值网络、状态价值网络组成,并且为每个神经网络定义合适的损失函数,实现集中训练和分散执行的优势策略^[20]。

2 问题陈述

针对在线网约车平台管理大量PDP问题,本文对空闲车辆和订单进行高效匹配。PDP问题属于经典网约车管理问题的一个变种^[21]。网约车订单派送示意图如图2所示。本文使用四边形网格表示地图,地图中的节点对应不同路网的交叉点,它们之间的边对应于不同的道路。车辆和订单随机出现在每个网格上,每条道路上都有相应的成本,该成本对应一辆车穿过道路的时间。成本包括不同交通条件在内的因素,如天气、节假日等,由环境决定,不能人为的定义。

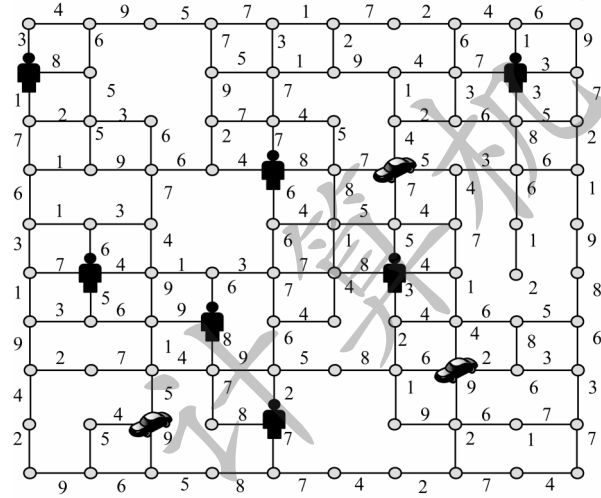


图2 网约车订单派送示意图

Fig.2 Schematic diagram of online car-hailing order dispatch

本文将网约车订单派送问题建模成MDP。智能体是从驾驶员的角度定义。完整的行程转换包括订单派送、订单接受和完成。网约车平台对空闲司机进行订单派送,司机接受订单并前往乘客初始位置,把乘客带到目的地,完成订单。司机在行程转换中立即获得奖励,即路费。本文把司机的一次闲置运动视为一次零奖励运动,定义MDP的关键要素 $G=(N,S,A,P,R,\gamma)$,分别表示车辆的数量、状态集合、联合动作空间、状态转移概率、奖励函数、折扣因子。

1) 状态 S_t 。状态输入表示为一个三元素元组,即 $S=(l,t,1)$ 元组中的元素,分别表示车辆的起始经纬度、时间、接单成功。

2) 动作 a_t 。对司机的一段特定行程进行分配,它简单由行程起始地点和下车时间定义。设当前 $S_0=(l_0,t_0,0)$ 为分配行程时司机位置、时间和车辆空闲, $S_1=(l_1,t_1,1)$ 为上车地点、时间和接单成功。因此,动作 $a=(l,t)$ 。所有符合条件动作的空间用 A

表示。

3) 奖励 r 。奖励函数在强化学习中非常重要,它在很大程度上决定了优化的方向。网约车通过接受订单到将乘客送到目的地完成订单,从而获得奖励。本文设计一个与每个订单的花费成比例的奖励函数,在每个单位时间步长中即时奖励序列的总和表示为 $R_t=r_{t+1}+\gamma r_{t+2}+\dots+\gamma^{k-t}r_{t+k}$ 。

4) 动作价值函数 $Q(s,a)$ 。司机采取动作时将获得累计奖励,直到一个时间步长结束。 $Q(s,a)=E\left[\sum_{t=0}^T \gamma^t R(S_t, A_t) | S_0=0, A_0=0\right]$ 。

5) 策略 $\pi(a|s)$ 。 $\pi(a|s)$ 是一个将状态映射到动作空间或特定动作上的分布策略。学习型 $Q(s,a)$ 的贪婪策略由 $\pi(s)=\operatorname{argmax} Q(s,a)$ 得出。

6) 状态值函数 $V(s)$ 。如果司机从开始遵循策略,将获得预期累计奖励,直到时间步长结束。策略 π 假设使用了 Q 函数的贪心策略,状态值 $V(s)=Q(s,\pi(s))=\max_{a \in A} Q(s,a)$ 。

3 基于联合 Q 值分解框架

网约车平台要实现高效的订单匹配,需要对外部环境有一个基本认识,主要包括订单的位置信息、时间等,综合考虑车辆当前状态的即时奖励和下一个状态的长期价值,通过最优策略函数选择最佳动作。VFD架构的核心思想是将原有的联合动作值函数转化为拥有最优动作的新函数。在 t 时刻,某区域的车辆最优联合动作相当于每辆车的最佳动作集合,在集中训练阶段易于因式分解的分布式决策任务的执行。

对于联合动作价值函数 $Q_j: S^N \times A^N \rightarrow R$,其中 $a \in A^N$ 是一个历史联合动作,如式(1)所示:

$$\operatorname{argmax}_a Q_j(s,a) = \begin{pmatrix} \operatorname{argmax}_{a_1} Q_1(s_1, a_1) \\ \vdots \\ \operatorname{argmax}_{a_N} Q_N(s_N, a_N) \end{pmatrix} \quad (1)$$

在式(1)中,存在独立动作函数 $[Q_i: S \times A \rightarrow R]_{i=1}^N$, $Q_j(s,a)$ 被 $Q_i(s_i, a_i)$ 分解或者 Q_i 是 Q_j 的因子,确保 $Q_j(s,a)$ 上的 argmax 函数得到的联合动作 a 与每个 $Q_i(s_i, a_i)$ 上 argmax 函数得到的独立动作 $[a_1, a_2, \dots, a_N]$ 一致,即每辆车的独立最优动作 $\operatorname{argmax}_{a_i} Q_i(s_i, a_i)$ 是联合最优动作 $\operatorname{argmax}_a Q_j(s,a)$ 的一部分。

3.1 整体结构

VFD框架克服现有方法中联合动作值函数分解的可加性和单调性的约束,能够分解所有可分解的任务。联合 Q 值分解函数的框架如图3所示,该框架由3个独立的网络构成:1)独立动作价值网络,主要是通过神经网络获得在 t 时刻某区域内每辆车的

动作值 $Q_i(s_i, a_i)$; 2) 联合动作价值网络, 主要是通过隐藏特征 $\sum_i h_{Q_i}(s_i, a_i)$ 得到在 t 时刻某区域内车辆的联合动作值 $Q_j(s, a)$; 3) 状态价值网络, 用于弥补联合动作价值网络学习到的动作值 $Q_j(s, a)$, 并与通过独

立动作价值网络得到所有车辆动作值之和 $Q'_j(s, a)$ 的差距。通过对 3 个神经网络进行集中训练, 每辆车在分散执行期间使用自己因式分解的独立价值函数 Q_i 采取动作。

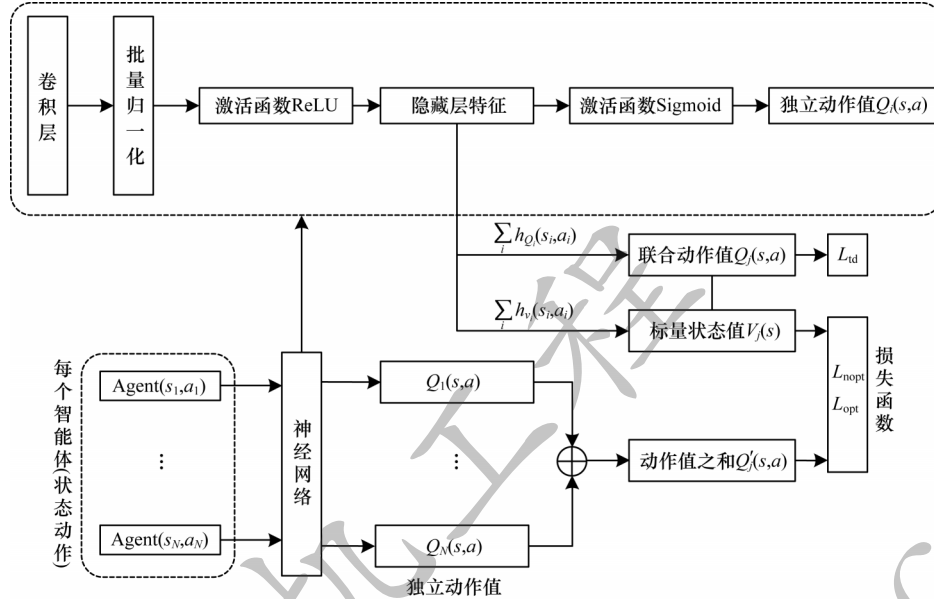


图3 联合Q值函数分解的框架

Fig.3 Framework of joint Q-value function decomposition

3.1.1 独立动作价值网络

对于每辆车, 独立动作价值网络根据其自身的历史观察 s_i 输入, 并将产生的动作值 $Q_i(s_i, a_i)$ 作为输出。通过计算给定的动作值来确定动作, 如式(2)所示:

$$Q'_j(s, a) = \sum_{i=1}^N Q_i(s_i, a_i) \quad (2)$$

其中: Q'_j 为所有车辆的动作值之和; $Q_i(s_i, a_i)$ 为每辆车的动作值。

3.1.2 联合动作价值网络

联合动作价值网络的 Q_j 值用于近似所有车辆的动作值之和 Q'_j , 将车辆所选动作作为输入, 生成所选动作的 Q 值并作为输出。为了提高网络的可扩展性和训练效率, 联合动作价值网络的设计是通过所有车辆独立动作价值网络确定的动作来更新联合动作价值网络。联合动作空间 N 是复杂的, 随着车辆数的增加, 寻找最优动作能提高复杂度, 而独立动作价值网络通过线性时间的个体最优动作的分散策略, 获得最优动作。联合动作价值网络共享独立动作价值网络的底层参数, 并且结合独立动作价值网络的隐藏特征 $\sum_i h_{Q_i}(s_i, a_i)$, 这些共享的参数可用于扩展训练。

3.1.3 状态价值网络

状态价值网络通过计算标量状态值 V_j , 用于弥

补 $Q'_j(s, a) = \sum_{i=1}^N Q_i(s_i, a_i)$ 与真实学习到的 $Q_j(s, a)$ 之间的差距, 如果没有状态价值网络, 部分可观测性将限制 Q'_j 的复杂性, 状态值与给定状态 s 的选定动作无关。因此, 状态价值网络不利于动作的选择, 而是用于计算式(3)和式(4)的损失值。与联合动作价值网络相同, 使用单个网络的组合隐藏特征 $\sum_i h_{V_i}(s_i, a_i)$ 作为状态价值网络的输入, 以提高空间扩展性。

定理1 联合动作价值函数 $Q_j(s, a)$ 被 $Q_i(s_i, a_i)$

分解, 如式(3)和式(4)所示:

$$\sum_{i=1}^N Q_i(s_i, a_i) - Q_j(s, a) + V_j(s) = 0, a = \bar{a} \quad (3)$$

$$\sum_{i=1}^N Q_i(s_i, a_i) - Q_j(s, a) + V_j(s) \geq 0, a \neq \bar{a} \quad (4)$$

其中: \bar{a} 表示局部车辆最优动作 $\arg\max_{a_i} Q_i(s_i, a_i)$; a 表示实际选择的动作。由于 $\bar{a} = \arg\max_{a_i} Q_i(s_i, a_i)$ 是局部车辆最优动作, 因此只需要 $Q_j(s, \bar{a}) \geq Q_j(s, a), a \neq \bar{a}$, 使所有局部最优动作 \bar{a} 的值大于其他动作 a 的值, 那么此时的 \bar{a} 就是全局最优动作。因为累加得到的 $Q'_j(s, a)$ 与真实学到的 $Q_j(s, a)$ 之间有差距, 所以用标量 V_j 进行平衡。当 $a = \bar{a}$, 执行式(3), $a \neq \bar{a}$ 执行式(4)。

3.2 损失函数

集中训练有 2 个主要目标: 训练联合动作价值

函数,计算真正的动作价值;转化后的动作价值函数 Q_j 应近似联合动作价值函数,它们的最优动作是等价的。本文使用DDQN方法来更新网络^[22],设计全局损失函数,将3个损失函数加权组合,如式(5)所示:

$$L(s, a, r, s'; \theta) = L_{td} + \lambda_{opt} L_{opt} + \lambda_{nopt} L_{nopt} \quad (5)$$

其中: r 表示在智能体观测环境执行动作 a 后,状态 s 转换到下一状态 s' 所获得奖励; L_{td} 表示估算实际动作价值的损失函数,通过学习 Q_j 使误差最小化; L_{nopt} 和 L_{opt} 表示满足式(3)和式(4)的因式分解损失函数, L_{nopt} 的作用是验证在样本中选择的动作是否满足式(4),则进一步确认 L_{opt} 得到的最优局部动作是否满足式(3)。因此,本文通过定义损失函数来实现式(3)和式(4),根据网络对样本中所采取的动作是否满足式(3)或式(4),但是这样验证式(3)和式(4)所得到的动作将需要过多的样本。由于在训练中很少采取最优动作,因此本文VFD框架的目标是学习 Q'_j 和 V_j 对给定的 Q_j 进行因式分解,在使用 L_{nopt} 和 L_{opt} 进行学习时,通过修正 Q_j 使学习更加稳定。 λ_{opt} 和 λ_{nopt} 是2个损失的权重常数,损失函数如式(6)~式(8)所示:

$$L_{td} = (Q_j(s, a) - y^{DDQN}(r, s'; \theta))^2 \quad (6)$$

$$L_{opt} = (Q'_j(s, \bar{a}) - Q_j(s, \bar{a}) + V_j(s))^2 \quad (7)$$

$$L_{nopt} = (\min[Q'_j(s, a) - Q_j(s, a) + V_j(s), 0])^2 \quad (8)$$

在上述3个损失函数中, L_{td} 使用标准时序差分TD-error来更新需学习的联合动作值 $Q_j(s, a)$, L_{opt} 和 L_{nopt} 通过 $Q_j(s, a)$ 来引导独立动作值之和 $Q'_j(s, a)$ 与标量状态值 $V_j(s)$ 的更新。

根据式(4),在ODDRL中通过独立动作价值函数 Q'_j 和状态价值函数 V_j 来更新联合动作值 Q_j ,导致神经网络无法准确地构建 Q_j 因式分解函数。式(4)条件可能会影响非最优动作,从而降低训练过程的稳定性或收敛速度。在此基础上,本文提出另一种方法LF-ODDRL,如定理2所示。

定理2 在定理1成立的条件下,用式(9)替换式(4),如果 $a \neq \bar{a}$,则:

$$\min_{a_i \in A} [Q'_j(s, a_i, a_{-i}) - Q_j(s, a_i, a_{-i}) + V_j(s)] = 0 \quad (9)$$

其中: $a_{-i} = (a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$ 。式(9)设置某些动作中的 $Q'_j(s, a) - Q_j(s, a) + V_j(s)$ 值为零,在现实情况中,式(9)不可能每个动作值都为0,可能非最优动作值 $Q'_j(s, a)$ 与最优动作值 $Q_j(s, \bar{a})$ 相差较大,导致实际学习过程不稳定。然而,式(9)是减小 $Q'_j(s, a)$ 与 $Q_j(s, \bar{a})$ 的差距,使学习更加稳定,因此,式(9)的实用性更强,更新后的 L_{nopt} 损失函数如式(10)所示:

$$L_{nopt} = \frac{1}{N} \sum_{i=1}^N (\min [Q'_j(s, a) - Q_j(s, a) + V_j(s), 0])^2 \quad (10)$$

ODDRL算法流程如下:

算法1 ODDRL算法

输入 每辆车的观测状态

输出 每辆车的调度动作

1. 初始化经验回放池D
2. 初始化 $[Q_i], [Q_j]$ 和参数 θ
3. 初始化目标参数 $\theta^- = \theta$
4. for episode = 1 to M do
5. 观察每辆车得到状态 $s^0 = [o(s^i, i)]^N$
6. for t = 1 to T do
7. 有概率地选择一个随机动作 a_t^i
8. 否则每辆车的 $a_t^i = \arg\max_{a_i} Q_i(s_t^i, a_i)$
9. 执行动作 a_t^i ,转换到下一个状态和得到收益 (s^{t+1}, r^t)
10. 存储数据 (s^t, a^t, r^t, s^{t+1}) 到回放池D中
11. 随机在回放池D中进行采样 (s, a, r, s')
12. $y^{DDQN} = r + \gamma Q_j \left(s', \left\{ \left[\arg\max_{a_i} Q'_i(s', a_i; \theta) \right]_{i=1}^N; \theta \right\} \right)$
13. if ODDRL通过式(5)~式(8)更新参数 θ :
if LF-ODDRL通过式(5)~式(7),式(10)更新参数 θ :
14. 更新目标网络参数 $\theta^- = \theta$
15. end for
16. end for

4 实验结果与分析

4.1 模拟平台

与有监督学习问题相比,多智能体强化学习的交互性给训练和评估带来更多的困难。常见的解决方案是建立环境模拟器^[23-24]。模拟器作为多智能体强化学习方法的训练环境,以及它们的评估器,是整个训练和学习的基础。文献[25]通过训练真实的历史数据产生模拟订单。在本文实验中,使用文献[26]的模拟平台模拟订单的生成,实现分配订单的过程。根据每辆网约车在平台上第一次的位置和时间进行初始化,驾驶员随后的动作由模拟器决定,如进行空闲移动或者离线/在线操作。模拟器经过仔细校准,对比模拟数据与真实数据,使两者的误差在允许范围内。

在基于网格模拟器中,将城市定义为图2所示的一张四边形网格的地图。在每个时间步,模拟器提供一个环境、一组空闲车辆和可用的订单。所有模拟订单与真实订单都具有相同的属性。为了提高模拟器的效率,每个乘客通过相同的网络同时计算动作值。因此,这个网络的输出是一个矩阵,其行表示乘客,其列表示每个乘客可以匹配的车辆。 (i, j) 的矩阵值是第 i 名乘客和第 j 辆车匹配的动作值,每个乘客与列表中 Q 值最大的车配对,使得奖励最大化,并且乘客的等待时间最小化。模拟器训练过程示意图如图4所示。

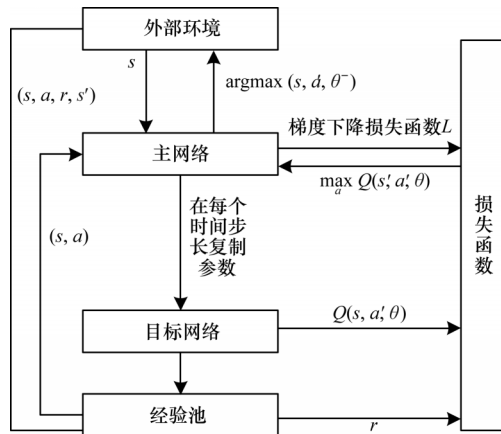


图 4 模拟器训练过程示意图

Fig.4 Schematic diagram of simulator training process

4.2 实验结果

为了分析方法的性能,本文采用 Random、Greedy、DQN 和 QMIX 作为基线方法。

1)Random:模拟没有任何订单派送的场景,只在每个时间步长随机分配闲置订单,并随机派送给网约车。

2)Greedy:贪婪方法被作为基线方法,并将其与各种强化学习方法进行对比。贪婪方法遵循先到先得(FCFS)策略。因此,较早要求用车的乘客会获得更高的优先权。每个乘客根据距离(曼哈顿距离)配到一辆车。

3)DQN:文献[27]提出一种基于 Q 学习网络的动作价值函数近似方法,该网络的参数化是由一个具有 4 个隐层的多层感知机(MLP)组成,隐层之间采用 ReLU 激活函数激活,并对 Q 网络的最终线性输出进行转换,使得车辆与环境之间进行局部交互,从

而捕获全局的动态需求与供应变化关系。

4)QMIX:考虑了全局状态,将联合动作价值网络函数分解为独立价值函数并进行分散执行^[28],使联合动作值最大化的联合动作等于使独立动作值最大化的一组动作。全局状态动作值允许每辆车贪婪地选择与它们独立动作值相关的动作。

为了测试 ODDRL 和 LF-ODDRL 方法在订单派送中的有效性和鲁棒性,本文在不同的网格、乘客数量和车辆数的情况下进行 3 组实验,所提方法在整体上都优于其他方法。如 QMIX 方法能够分解联合动作价值函数,确保全局最优动作和局部最优动作的一致性,使得动作值最大化,但是通过神经网络学习全局与局部之间的关系,确保全局动作值与局部动作值的单调性,仅解决了小部分可分解任务的问题。因此,该方法存在一定的局限性。本文方法克服了联合动作值分解可加性和单调性的约束,以确保联合动作和独立动作是相同的,从而提高学习效率和扩展性。

本文将网约车和乘客数量设置为固定的,在 100×100 的网格上,使用把所有乘客送达目的地所持续的总时间作为评判指标,6 种方法的总时间对比如表 1 所示。 P 和 C 分别表示在固定人车网格中每回合初始的乘客数量和车辆数量,当 $P=7, C=2$ 时,ODDRL 和 LF-ODDRL 两种方法相对于 QMIX 接送乘客持续总时间分别减少了 6% 和 18%,当有大量车辆和乘客时,如 $P=25, C=20$,持续总时间分别减少了 10% 和 18%。实验结果表明,本文方法能有效解决网约车不足和充分问题。

表 1 在 100×100 网格上不同方法接送乘客的总时间对比

Table 1 Total time for picking up passengers comparison among different methods on 100×100 grid

方法	总时间/s					
	$P=7, C=2$	$P=10, C=10$	$P=11, C=13$	$P=9, C=4$	$P=10, C=2$	$P=25, C=20$
Random	3 386.248	2 210.870	2 089.870	2 958.861	4 644.590	2 962.790
Greedy	3 526.959	2 208.550	2 089.630	3 072.806	4 934.910	3 173.540
DQN	3 402.053	2 099.360	2 058.920	2 852.752	4 965.360	2 752.360
QMIX	3 184.546	2 056.780	1 956.240	2 752.748	4 369.860	2 762.240
ODDRL	3 000.215	1 812.302	1 729.297	2 623.466	4 277.643	2 499.821
LF-ODDRL	2 600.150	1 694.914	1 603.942	2 520.335	4 155.790	2 272.451

为进一步衡量方法的可伸缩性,在 10×10 和 500×500 网格上,不同方法接送乘客持续总时间随网约车和乘客数量的变化曲线分别如图 5 和图 6 所示。其中 Random、Greedy、DQN 方法都是针对单个智能体,并没有联合所有智能体集中训练和分散执行的策略,因此,其结果并不理想。QMIX 方法虽然考虑集中训练和分散执行的策略,但是需联合动作值和独立动作值呈现

单调性,阻碍了对非单调性结构动作值函数的学习。而本文方法却弥补了这些缺点,不仅联合车辆共同学习,还针对不同规模的网格构建 ReLU 激活函数和训练订单派送的策略 Q 网络,并且设计新的损失函数来缩小 $Q'_j(s, a)$ 与 $Q_j(s, a)$ 的差距。实验结果表明,ODDRL 与 LF-ODDRL 都优于其他 4 种基线方法,说明本文方法具有较优的扩展性。

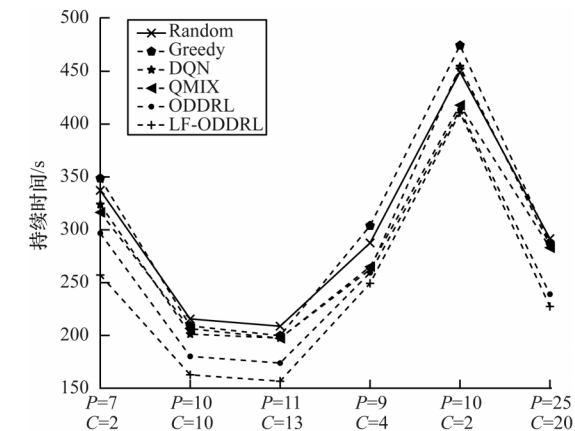


图5 在10×10网格上不同方法接送乘客的总时间对比

Fig.5 Total time for picking up passengers comparison among different methods on 10×10 grid

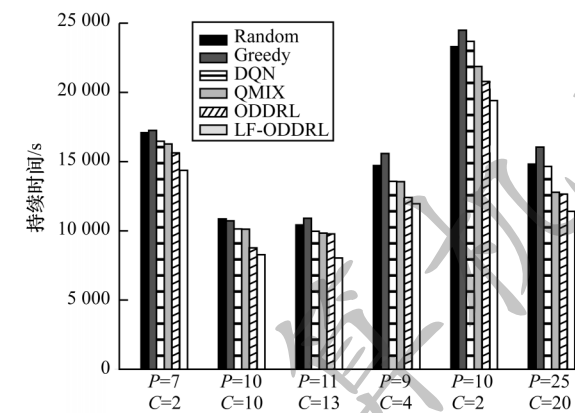


图6 在500×500网格上不同方法接送乘客的总时间对比

Fig.6 Total time for picking up passengers comparison among different methods on 500×500 grid

在10×10和500×500的网格上,本文给网约车和乘客数量分别设置域值,并进行训练和测试,接送乘客所持续的总时间如表2所示。在500×500的网格上,当最大乘客数和车辆数都为20时,ODDRL与LF-ODDRL相对于QMIX接送乘客所持续的总时间分别减少了2%和12%,说明VFD在应对可变的乘客数和车辆数时,具有更强的学习能力,以更好适应可变的环境。

表2 不同方法接送乘客的总时间对比		
Table 2 Total time for picking up passenges comparison among different methods		
方法	总时间/s	
	10×10 网格 $P_{\max}=10,C_{\max}=10$	500×500 网格 $P_{\max}=20,C_{\max}=20$
Random	209.030	13 700.140
Greedy	201.850	13 871.070
DQN	199.430	13 462.820
QMIX	195.660	12 812.790
ODDRL	181.660	12 530.460
LF-ODDRL	177.699	11 188.454

为验证本文设计动作搜索损失函数的有效性,本文在10×10网格 $P=7,C=2$ 的条件下进行实验。不同方法的损失值对比如图7所示。本文设计的损失函数是从当前的Q网络中找出最大Q值对应的动作,利用这个选择的动作在目标网络中计算目标Q值。从图7可以看出,LF-ODDRL的收敛速度和效果都优于ODDRL方法。

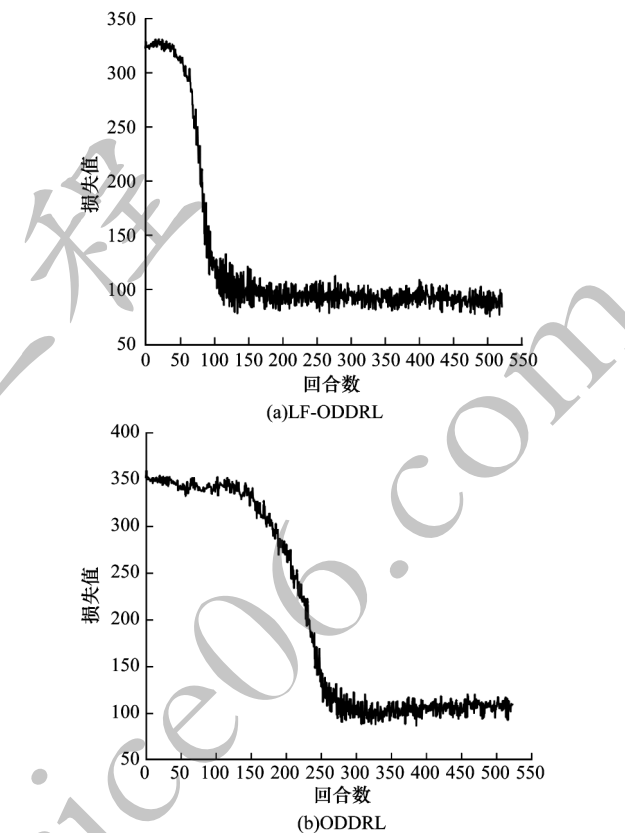


图7 不同方法的损失值对比

Fig.7 Loss values comparison among different methods

5 结束语

本文基于VFD框架提出多智能体强化学习方法ODDRL和LF-ODDRL,用于解决PDP问题。将订单派送建模为一个分布式决策问题,把初始的联合动作值函数分解为多个独立的值函数,使联合动作值函数中的动作与独立动作值函数中的动作相一致。将每辆车以分布式的方式执行动作,通过集中训练和分散执行的方式确定车辆与订单之间的最佳匹配关系,达到优化平台长期运作效率的目的,同时最大程度缩短接送时间,提高乘客舒适度。实验结果表明,相比Random、Greedy、DQN等方法,ODDRL和LF-ODDRL能够有效缩短接送时间,在不同规模网格上具有较优的扩展性。后续将原始车辆GPS数据与乘客的个人偏好及其目的地距离相结合,最大化缩短接送时间,从而提高乘客和司机的出行效率。

参考文献

- [1] ZHOU M, JIN J R, ZHANG W N, et al. Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching[C]//Proceedings of the 28th International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2019: 2645-2653.
- [2] VERMA T, VARAKANTHAM P, KRAUS S, et al. Augmenting decisions of taxi drivers through reinforcement learning for improving revenues[EB/OL]. [2021-10-28]. https://www.researchgate.net/profile/Tanvi-Verma-3/publication/324963776_Augmenting_Decisions_of_Taxi_Drivers_through_Reinforcement_Learning_for_Improving_Revenues/links/5aed037b458515f59982eccf/Augmenting-Decisions-of-Taxi-Drivers-through-Reinforcement-Learning-for-Improving-Revenues.pdf.
- [3] JIAO Y, TANG X C, QIN Z W, et al. Real-world ride-hailing vehicle repositioning using deep reinforcement learning[J]. Transportation Research Part C: Emerging Technologies, 2021, 130: 103289.
- [4] WALDY J, HOONG C L. Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers[C]//Proceedings of International Conference on Automated Planning and Scheduling. [S. l.]: AAAI Press, 2020: 394-402.
- [5] ZHENG H Y, WU J. Online to offline business: urban taxi dispatching with passenger-driver matching stability[C]//Proceedings of the 37th International Conference on Distributed Computing Systems. Washington D. C., USA: IEEE Press, 2017: 816-825.
- [6] ZHANG R, PAVONE M. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective[J]. International Journal of Robotics Research, 2014, 35(1/2/3): 186-203.
- [7] MA Y N, LI J W, CAO Z G, et al. Learning to iteratively solve routing problems with dual-aspect collaborative Transformer[EB/OL]. [2021-10-28]. <https://arxiv.org/abs/2110.02544>.
- [8] WU Y X, SONG W, CAO Z G, et al. Learning improvement heuristics for solving routing problems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(9): 5057-5069.
- [9] ZHANG L Y, HU T, MIN Y, et al. A taxi order dispatch model based on combinatorial optimization[C]//Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2017: 2151-2159.
- [10] WEI Z Q, CHENG X T, YAO J, et al. Ride-hailing order dispatching at DiDi via reinforcement learning[J]. INFORMS Journal on Applied Analytics, 2020, 50(5): 272-286.
- [11] LI J W, XIN L, CAO Z G, et al. Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(3): 2306-2315.
- [12] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. Cambridge, USA: MIT Press, 2018.
- [13] XU Z, LI Z X, GUAN Q W, et al. Large-scale order dispatch in on-demand ride-hailing platforms: a learning and planning approach[C]//Proceedings of the 24th International Conference on Knowledge Discovery & Data Mining. New York, USA: ACM Press, 2018: 905-913.
- [14] LI J W, MA Y N, GAO R Z, et al. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem[J]. IEEE Transactions on Cybernetics, 2021, 99: 1-10.
- [15] 邱月, 郑柏通, 蔡超. 多约束复杂环境下 UAV 航迹规划策略自学习方法[J]. 计算机工程, 2021, 47(5): 44-51.
- [16] QIU Y, ZHENG B T, CAI C. Self-learning method of UAV track planning strategy in complex environment with multiple constraints[J]. Computer Engineering, 2021, 47(5): 44-51. (in Chinese)
- [17] WANG Z D, QIN Z W, TANG X C, et al. Deep reinforcement learning with knowledge transfer for online rides order dispatching[C]//Proceedings of International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2018: 617-626.
- [18] YANG Y D, LUO R, LIN M N, et al. Mean field multi-agent reinforcement learning[EB/OL]. [2021-10-28]. <https://arxiv.org/pdf/1802.05438.pdf>.
- [19] AL-ABBASI A O, GHOSH A, AGGARWAL V. DeepPool: distributed model-free algorithm for ride-sharing using deep reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(12): 4714-4727.
- [20] JIAO Y, TANG X C, QIN Z W, et al. Real-world ride-hailing vehicle repositioning using deep reinforcement learning[J]. Transportation Research Part C: Emerging Technologies, 2021, 130: 103289.
- [21] SON K, KIM D, KANG W J, et al. Qtran: learning to factorize with transformation for cooperative multi agent reinforcement learning[EB/OL]. [2021-10-28]. <https://arxiv.org/abs/1905.05408>.
- [22] ZHANG W Q, WANG Q, LI J J, et al. Dynamic fleet management with rewriting deep reinforcement learning[J]. IEEE Access, 2020, 8: 143333-143341.
- [23] 雷捷维, 王嘉吻, 任航, 等. 基于 Expectimax 搜索与 Double DQN 的非完备信息博弈算法[J]. 计算机工程, 2021, 47(3): 304-310, 320.
- [24] LEI J W, WANG J Y, REN H, et al. Incomplete information game algorithm based on Expectimax search and Double DQN[J]. Computer Engineering, 2021, 47(3): 304-310, 320. (in Chinese)
- [25] MACIEJEWSKI M, NAGEL K. The influence of multi-agent cooperation on the efficiency of taxi dispatching[C]//Proceedings of International Conference on Parallel Processing and Applied Mathematics. Berlin, Germany: Springer, 2014: 751-760.
- [26] WEI C, WANG Y H, YAN X D, et al. Look-ahead insertion policy for a shared-taxi system based on reinforcement learning[J]. IEEE Access, 2017, 6: 5716-5726.
- [27] JIN J, ZHOU M, ZHANG W, et al. Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms[C]//Proceedings of the 28th International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2019: 1983-1992.

(下转第 311 页)

(上接第 303 页)

[26] LIMA O D, SHAH H, CHU T S, et al. Efficient ridesharing dispatch using multi agent reinforcement learning[EB/OL]. [2021-10-28]. <https://arxiv.org/abs/2006.10897>.

[27] LI M N, QIN Z W, JIAO Y, et al. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning [C]//Proceedings of Efficient Ridesharing Order Dispatching with Mean Field Multi-Agent Reinforcement Learning. New York, USA: ACM Press, 2019:983-994.

[28] RASHID T, SAMVELYAN M, SCHROEDER C, et al. QMIX: monotonic value function factorization for deep multi agent reinforcement learning[EB/OL]. [2021-10-28]. <https://arxiv.org/pdf/1803.11485.pdf>.

编辑 薛晋栋