控制与决策
Control and Decision

考虑工人路径的多智能体强化学习空间众包任务分配方法

纪苗苗, 吴志彬

引用本文:
纪苗苗,吴志彬. 考虑工人路径的多智能体强化学习空间众包任务分配方法[J]. 控制与决策, 2024, 39(1): 319−326.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2022.1319

考虑工人路径的多智能体强化学习空间众包任务分配方法

纪苗苗，吴志彬†

(四川大学 商学院，成都 610065)

摘 要: 针对工人和任务进行匹配是空间众包研究的核心问题之一,但已有的方法通常会忽略工人路径对任务分配结果产生的影响. 传统的任务分配方法存在计算速度慢、适用范围小和协作效果不突出等问题. 对此,从空间众包平台的角度出发研究面向路网的空间众包任务分配问题,以任务完成时间最短为目标,提出考虑工人路径规划的基于多智能体强化学习的QMIX-A* 算法,缩短任务的平均完成时间,进而提高用户的满意度. 大量的数值仿真研究验证了QMIX-A* 的有效性和稳定性,为空间众包服务平台的任务分配与路径优化策略的选择提供决策支持.

关键词: 多智能体；强化学习；空间众包；任务分配；路径优化；路网

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2022.1319

引用格式: 纪苗苗,吴志彬. 考虑工人路径的多智能体强化学习空间众包任务分配方法 [J]. 控制与决策, 2024, 39(1): 319-326.

A multi-agent reinforcement learning algorithm for spatial crowdsourcing task assignments considering workers' path

JI Miao-miao, WU Zhi-bin†

(Business School，Sichuan University，Chengdu 610065，China)

Abstract: Matching tasks and workers is one of the core problems in spatial crowdsourcing research, but the impact of path planning of workers on task allocation results is usually ignored in the existing literature. There are problems with traditional task assignment methods including slow computing speed, small application scope, and unremarkable collaboration effect. From the perspective of a spatial crowdsourcing platform, this research is oriented toward the spatial crowdsourcing task assignment problem on the road networks and puts forward a QMIX-A* algorithm based on multi-agent reinforcement learning considering workers' path planning. The proposed approach with the minimum completion time of tasks as the objective can shorten the tasks' average completion time, thereby improving users' satisfaction. The effectiveness and stability of the QMIX-A* are verified by a large number of simulation studies. The results of the research can provide decision support for the task allocation and path optimization strategy selection of spatial crowdsourcing service platforms.

Keywords: multi-agent；reinforcement learning；spatial crowdsourcing；task assignment；path planning；road network

0 引 言

随着移动互联网的发展、基于GPS的智能设备的普及以及共享经济新概念的提出,空间众包逐渐吸引了来自工业界和学术界的广泛关注. Uber、UU跑腿、货拉拉、滴嗒出行、58到家等空间众包应用的快速发展,也使得众包商业模式在社会治理、交通管理、灾情监控等领域发挥了重大作用. 用户在众包平台上发布带有空间属性的任务,众包平台根据不同的优化目标,如最大化任务分配数量或工人收益、最小化工人行程代价等,指派合适的工人前往指定地点完成任务. Tong等[1]对空间众包任务给出了一般定义:给定一组任务和一组工人,在满足空间约束、时间约束和其他约束的前提下,为了特定目标在工人与任务之间作出安排的过程.

空间众包问题按算法模型可分为匹配模型和规划模型,前者着重任务匹配,后者则强调路径规划. 但是,现有的空间众包问题大多只考虑到工人和任务,对于工人执行任务时的路径规划研究则较少,任务分

收稿日期: 2022-07-23；**录用日期:** 2022-09-20.

基金项目: 国家自然科学基金面上项目(71971148)；中央高校基本科研业务费专项资金项目(SXYPY202103).

†通讯作者. E-mail: zhibinwu@scu.edu.cn.

配时通常只考虑距离范围,未考虑路径和基于路径产生的移动成本对分配结果的影响. 潘庆先等[2]提出了基于自适应阈值的禁忌搜索算法,在考虑众包工人的路径规划问题和移动成本情况下,通过合理匹配使得区域内的众包任务能在最短的时间内得到分配. 戴韬等[3]提出了由遗传算法和贪心算法组成的双层算法,分别解决众包模式下的订单选择及订单执行路径问题. 余海燕等[4]针对众包配送平台中订单的实时性、时效性、配送员的自由性等特征,建立了以平均每单配送距离以及平均每单完成时间最小为目标的实时订单分配与路径优化模型.

优化问题的目标通常可以概括为最大效用和最小成本两大类. 为了实现这些目标,人们提出了一系列的精确和启发式算法. To 等[5]将静态匹配问题转化为最小最大加权二部匹配问题并利用 Hungarian 算法来获得精确解. 为了最大化任务分配数量,Deng 等[6]提出了动态规划算法、基于分支定界的算法等精确算法,并提出了几种剪枝策略来缩短算法的运行时间. 启发式算法在计算效率方面具备优势. 赵杨等[7]针对面向空间众包平台的多工作者多任务路径规划问题,以求解时间成本和路程成本最小的全局最优路径规划方案为目标,提出了基于改进狮群进化算法的路径规划方法. 吴腾宇等[8]研究了带有取送货的在线旅行商问题,针对需求点在不同网络上的情形分别设计了 TAIB 算法和 IGNORE 算法求解. 传统算法中的精确算法常常由于过高的时间复杂度而无法在实际中应用,而启发式算法的搜索机制和策略往往需要具备丰富的工程经验和专业知识,且容易陷入局部最优.

随着人工智能的迅速发展,机器学习算法被用来解决任务分配问题. 例如,Safran 等[9]使用基于类别的矩阵分解和 KNN 算法向工人推荐任务. 然而,由于需要历史经验以及难以适应变化的环境,有监督学习方法并未被广泛使用. 人工智能的热潮也促进了深度强化学习的研究和发展,结合深度学习的感知能力和强化学习的决策能力的深度强化学习算法被广泛应用于各个领域,如网约车派单优化[10]、车间动态调度[11]、交通信号灯智能控制[12]、无人机路径规划[13]和网络资源在线分配[14]等. 与传统基于模型和求解算法的方法相比,深度强化学习在处理复杂系统决策和控制问题方面具有优势. 通过高维感知输入学习使得各个智能体根据奖励或惩罚规则,结合所处的环境与状态,采取某种行动,并通过对记忆的不断训练与优化,从而达到最优的结果. 深度强化学习通过与环境交互来探索得到最优策略,为解决空间众包任务提供了可能. 例如,Liu 等[15]和 Shan 等[16]分别将强化学习用于众包中的工人调度和任务推荐,但是,这种智能体强化学习方法难以支持批处理和工人间的合作,因此性能受到限制. Jiang 等[17]首次以多智能体的视角对众包进行了全面总结,指出多智能体系统与众包系统关系密切且多主体方法可以为众包技术带来有效的提升. 受此启发,本文尝试在空间众包中使用多智能体强化学习进行任务分配. 多智能体深度强化学习是多智能体学习与深度强化学习的结合,通过多个分布式自运行的智能体与环境的交互,以及智能体之间的协调与平衡,实现共同的利益或特定的目标. 相比于单智能体强化学习,多智能体强化学习能更自然地分解和描述复杂的现实问题,也更具有扩展性.

本文以 QMIX 多智能体强化学习算法[18]为基础,将带取送货的路径规划问题抽象为经过必经点最短路径问题,使用 A* 算法[19]计算环境给予智能体的奖励,提出 QMIX-A* 算法. 为了解决多智能体系统的协同控制问题,QMIX-A* 算法采用中心式训练、分布式执行框架.

1 空间众包任务分配场景

本研究是从空间众包平台的角度出发解决空间众包任务匹配问题,目的是缩短某个时间段内所有任务的平均完成时间,而不是某个工人的任务完成时间. 图 1 是中心化空间众包智能匹配系统示意. 在某个时间段内,用户在空间众包平台上发布任务,平台接收信息包括用户任务的起点、终点位置信息等并同时获取所有工人的实时位置,平台将任务分配给相应的工人. 每个工人接受任务后需要到达任务起始位置,然后前往目标位置完成任务. 在该过程中,由于用户发布的任务具有实时性,每个任务请求都要完成且要求在尽可能短的时间内完成,即具有很强的时效

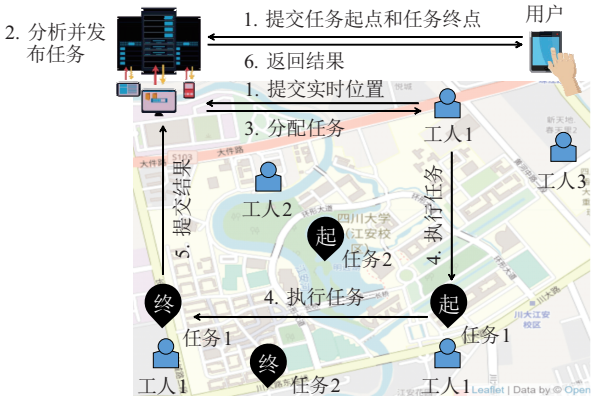


图 1 中心化空间众包智能匹配系统

性,且每个任务的取送地点一一匹配. 平台根据用户的任务需求和实时收集的工人位置信息,对工人和任务进行匹配.

本文通过一个无向带权图$G = \langle V, E, C\rangle$表示路网,路网示意如图2所示. 其中:每个节点$v \in V$表示一个实际位置点且坐标唯一确定;每条边$e = (v_i, v_j)$表示连接两个实际位置点的路段;每条边都与一个权重$c(e) \in C$关联,权重表示经过这条边的成本. 成本可以用距离或时间来衡量. 当工人速度已知时,距离和时间可以相互转换,因此本文将时间作为成本. 不妨设某时间段T内在G上有m个工人和n个任务. 工人可以同时匹配多项任务,并规定工人遵循先匹配先执行的顺序完成任务[20]. 若工人当前未被分配任务,则称为空闲工人;否则称为非空闲工人. 相应地,引入立即匹配与延时匹配的概念:当用户发布任务时,若匹配到的是空闲工人,则工人接受任务会立即赶往任务的起始位置执行任务,这种匹配方式被定义为立即匹配;若匹配到的工人是非空闲工人,则工人要完成正在执行的任务后再赶往任务的起始位置执行任务,这种匹配方式被定义为延时匹配. 本文要研究的问题是在为工人规划路径的同时,如何实时分配任务给合适的工人,使得给定时间段内任务的平均完成时间最短,进而提高用户的满意度.

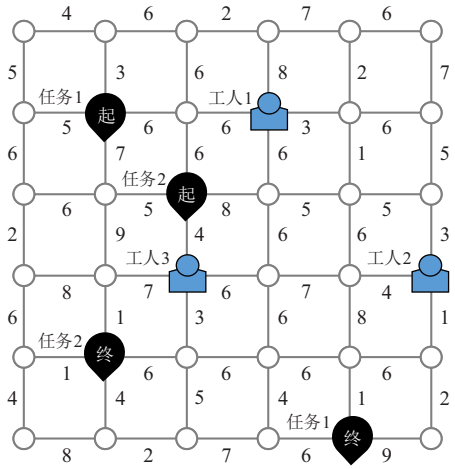


图 2　路网示意图

2　基于多智能体强化学习的空间众包任务分配方法

本文的空间众包场景可以刻画为一个完全协作的多智能体问题,并用马尔可夫模型来表示. 该模型由元组$\langle S, U, P, r, n, \gamma\rangle$组成,它描述的是智能体$a \in A \equiv \{1, 2, \ldots, n\}$基于环境状态$s \in S$选择合适的动作$u_a \in U$,$n$个独立动作构成联合动作$\boldsymbol{u} \in U^n$,以及获得奖励$r(s, \boldsymbol{u})$的过程. 其中:环境会根据状态转移函数$P : S \times U \to P$变化,深度学习使用动作价值函数$Q(s, \boldsymbol{u}) = E[R_T | S_T = s, |U_T = \boldsymbol{u}]$学习最优策略. 这里$R_T = \sum_{k=0}^{+\infty} \gamma^k r_{T+k}$表示累积奖励,$\gamma$是衰减因子且$\gamma \in [0, 1)$. 接下来,对本文的多智能体强化学习场景涉及到的基本要素包括智能体、动作、状态和奖励进行详细定义. 每个任务均被视为一个智能体,用$a \in A \equiv \{1, 2, \ldots, n\}$表示. 所有智能体拥有相同的动作空间,即所有工人的集合,$u_a \in U \equiv \{1, 2, \ldots, m\}$. 工人和任务的位置信息作为状态$s$输入,包括:1) 当前工人的位置信息$L_w = (l_x, l_y)$,$l_x$和$l_y$为工人当前位置的坐标;2) 任务起点和终点的位置信息$L_t = (p_x, p_y, d_x, d_y)$,$p_x$和$p_y$为任务起点的坐标,$d_x$和$d_y$为任务终点的坐标;3) 两个长度分别为$m$和$n$的二元指示向量$\boldsymbol{I}_w$和$\boldsymbol{I}_t$用来表示在时段$T$内工人和任务是否出现在路网$G$上,

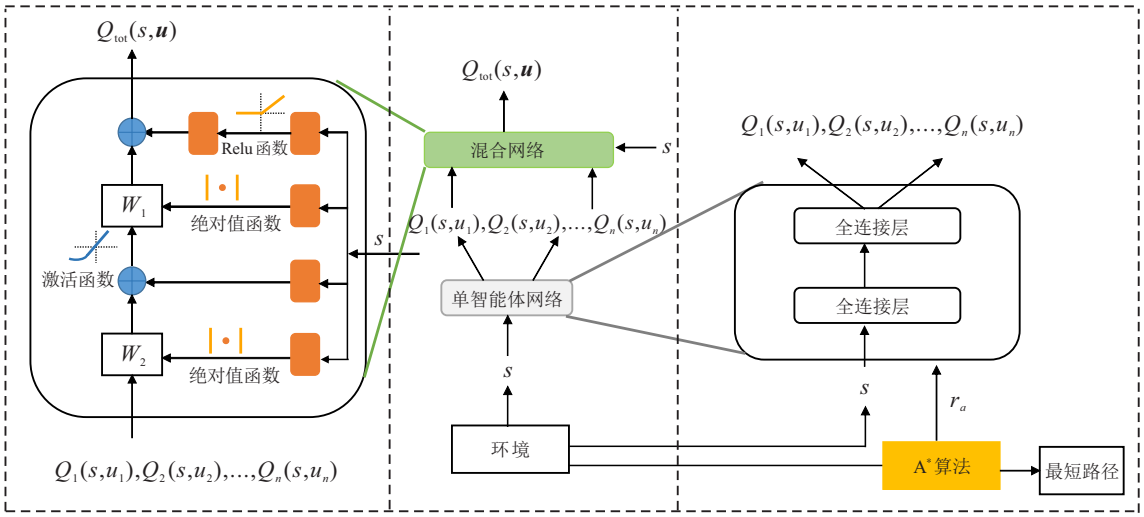


图 3　QMIX-A*算法结构示意

若时段T内,工人i或任务j在路网上,则I_{w_i}或I_{t_j}记为1,否则为0. 最终智能体基于环境的观察值即状态$s=(L_{w_1},L_{w_2},\dots,L_{w_m},\boldsymbol{I}_w,L_{t_1},L_{t_2},\dots,L_{t_n},\boldsymbol{I}_t)$,在时段$T$内,每个任务都会匹配给一个工人,每个任务完成将会获得奖励. 需要说明的是,本文考虑的是单步决策问题场景,因此$\gamma=0$.

QMIX-A*算法结构示意如图3所示,它包括单智能体网络和混合网络. 首先,收集各智能体的状态信息并将其输入到单智能体网络,通过状态s以及动作u_a得到动作价值函数Q_a,并将其输入到混合网络进行训练;然后,通过训练后的$Q_{\rm tot}(s,\boldsymbol{u})$分布式地获得每个智能体动作价值函数最大值所对应的动作,以解决高维状态空间问题;最后,实现多智能体系统的协同控制.

2.1 单智能体网络

单智能体网络是由两个全连接层组成的深度神经网络. 网络的输入是状态s,记为$D_{\rm input}\in R^{3m+5n}$. 为了提高算法的计算效率,所有任务的动作值函数均由一个单智能体网络计算. 因此,单智能体网络的输出是一个矩阵,它的行代表任务j,列代表每个任务可匹配的工人i. (i,j)处的矩阵值为第i个工人与第j个任务匹配时的动作值,输出$D_{\rm output}\in R^{mn}$. 当每个训练周期开始时,路网上的每个工人和每个任务的状态都是随机产生. 当工人和任务完成匹配后所有的任务都完成,则训练周期结束,不包含下一个状态,因此仅根据当前动作给予奖励. 奖励函数是环境给予智能体的反馈,它在很大程度上决定了优化的方向,因此奖励函数的计算非常重要.

2.2 A* 算 法

在单个智能体控制上,每个任务完成时的奖励为$(-1)\times$(任务完成时间),即

$$r_a=-\sum_{j=1}^{k_{w_i}}c(w_i,t_j). \tag{1}$$

其中:$c(w_i,t_j)$表示第i个工人执行第j个任务需要的的时间,k_{w_i}表示工人w_i接受的任务数量. 当$k_{w_i}=1$时,表示t_j匹配到的工人是空闲工人,属于即时匹配;当$k_{w_i}>1$时,表示t_j匹配的工人是非空闲工人,属于延迟匹配. 除了完成任务t_j的时间外,延迟匹配的奖励还应包括工人完成之前$k_{w_i}-1$个任务的等待时间. 考虑到每个智能体的路径规划属于取送货的问题,计算$c(w_i,t_j)$可分为:1) 工人从当前位置前往任务起始位置的时间;2) 工人从任务起始位置前往任务目标位置的时间. 为了简化起见,本文假设工人以相同的速度行进,时间的计算与距离正相关. 因此,任务的最短完成时间可以转化为通过固定点最短路径的求解问题,即分别计算起点(工人当前位置)到必经点(任务起始位置)以及必经点(任务起始位置)到终点(任务目标位置)的最短路径.

A*算法是静态路网求解最短路径经典且最有效的启发式搜索算法,具有计算过程简单、规划路径短等优点[21]. A*算法利用评价函数在每次搜索时都对所有可扩展节点进行评估,从而找到每次搜索的最佳扩展节点,直至找到目标节点. 评价函数可以如下表示:

$$f(v)=g(v)+h(v). \tag{2}$$

其中:$f(v)$表示从起点经由节点v到达终点的路径代价;$g(v)$表示当前节点v距离起点的实际路径代价;$h(v)$是启发函数,表示当前节点v距离终点的估计路径代价,通常基于曼哈顿距离、对角线距离和欧几里德距离进行计算,本文采用曼哈顿距离作为A*算法的启发函数. A*算法在每次搜索时,选择拥有$f(v)$最小值的节点作为下一次搜索的节点. $h(v)$的选择对于$f(v)$的计算具有关键性的作用,决定了A*算法的效率和速度. 由于中心化空间众包智能匹配系统的路网性质,通常工人不能全向移动,只能沿着水平方向或竖直方向行进. 因此,设置A*算法的搜索方向为四邻域的节点扩展方式[22],如图4所示.

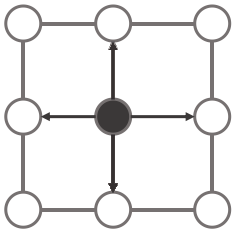


图 4 A*算法四邻域搜索方式

A*算法的实现步骤如下.

step 1: 创建Open_List列表和Close_List列表,分别用于存放待检测节点和已检测节点进行辅助搜索,并将起点v_s放入Open_List列表.

step 2: 检测Open_List列表是否为空. 若是,则因无法搜索到可行路线而算法结束;否则,选取Open_List列表中$f(v)$值最小的节点作为当前节点v,并将其从Open_List列表移到Close_List列表中.

step 3: 判断Close_List列表中的节点v是否为终点v_d. 若是,则利用回溯算法求解得出最短路径,结束算法;否则,以四邻域搜索方式继续进行扩展搜索子节点.

step 4: 判断子节点是否在Open_List列表中. 如果是,则需要计算该子节点的$f(v)$,并选取$f(v)$值最

小的节点保留;如果不是,则将该节点放入Open_List列表中,再计算$f(v)$.

step 5: 返回 step 2,继续循环以上步骤搜索,直至找到终点v_d,得出最短路径,结束算法.

2.3 混合网络

混合网络是一个前馈神经网络,它将各个智能体得到的$Q_1(s,u_1),\dots,Q_n(s,u_n)$作为输入,并将它们单调组合在一起,最终输出联合动作函数$Q_{\rm tot}(s,\boldsymbol{u})$. QMIX-A* 通过全局奖励$R_{\rm tol}$和学习联合动作函数$Q_{\rm tot}(s,\boldsymbol{u})$来促进智能体间的协调. 其中:$s$是对环境的观察,$\boldsymbol{u}$是所有任务的联合动作. 每个任务与$Q_a$值最大的工人按照先到先分配的原则进行匹配. QMIX-A* 算法中联合动作价值函数$Q_{\rm tot}(s,\boldsymbol{u})$与每个智能体的动作价值函数$Q_a$之间关系如下:

$$\arg\max_{\boldsymbol{u}} Q_{\rm tot}(s,\boldsymbol{u})=\begin{bmatrix}\arg\max\limits_{u_1} Q_1(s,u_1)\\ \arg\max\limits_{u_2} Q_2(s,u_2)\\ \vdots\\ \arg\max\limits_{u_n} Q_n(s,u_n)\end{bmatrix}. \tag{3}$$

QMIX-A* 利用集中式的训练得到分布式的策略. 当$Q_{\rm tot}(s,\boldsymbol{u})$取最大值等价于每个$Q_a$取最大值时,智能体可以通过学习到的$Q_a$选择最优动作. 此时需要满足

$$\frac{\partial Q_{\rm tot}}{\partial Q_a}\geqslant 0,\ \forall a\in\{1,2,\dots,n\}. \tag{4}$$

为了确保$Q_{\rm tot}(s,\boldsymbol{u})$与$Q_a$的单调性关系,混合网络的权重必须严格非负,但偏差不受非负约束. 混合网络的权重矩阵$\boldsymbol{W}_1$和$\boldsymbol{W}_2$均由一个线性层与一个绝对值激活函数构成的超网络生成,并用相同的方式生成偏差. Huber损失[23]处处可导且对离群值的敏感度较低,可以有效防止梯度爆炸. 当动作值的估计是有噪声时,使用Huber损失仍可以较快的速度更新参数并且能够保证模型更精确地得到全局最优值,即

$$H(x)=\begin{cases}\dfrac{1}{2}x^2,\ \forall|x|\leqslant 1;\\ |x|-\dfrac{1}{2},\ \forall|x|>1.\end{cases} \tag{5}$$

则最终损失函数为

$$L(\theta)=\frac{1}{B}\sum_{b=1}^{B}H(R_{\rm tot}^b-Q_{\rm tot}(s,\boldsymbol{u},\theta)). \tag{6}$$

其中:θ是QMIX-A* 网络的参数;B是批量大小;$R_{\rm tot}^b$是时间段T内第b个批量所有任务被完成时环境给予的全局奖励,即$\sum\limits_{a=1}^{a=n}r_a$.

3 仿真实验

3.1 实验设置

本文基于Python环境并使用Windows10操作系统,利用Pytorch构建多智能体强化学习模型,使用的处理器为1.90 GHz AMD Ryzen 7 5800U with Radeon Graphics,内存为16 GB. 为了验证QMIX-A* 算法的性能,采用3种比较方法作为基线方法:随机算法、贪婪算法和IQL-A* 算法.

1) 随机算法:随机算法模拟的是每个时间段内任务被随机分配给工人的场景.

2) 贪婪算法:它是一种基于规则的任务分配算法. 任务总是被分配给距离起始位置最近的工人,工人移动成本同样是基于A* 算法计算出的最短路径. 贪婪算法将作为衡量其他算法的基准算法,并将其与各种强化学习方法进行比较.

3) IQL-A* 算法:多指智能体强化学习算法IQL (independent Q-learning)算法[24]与A* 算法的结合,IQL算法是基于Q-learning网络的动作价值函数近似方法.

对于多智能体强化学习算法,设置ϵ-贪心策略的初始探索值为0.9,最终探索值设为0.05,衰变周期设为25 000. 总的训练周期为50 000,回放记忆缓存的大小设为8 000,批量的大小B设为128,使用学习率设为0.001的Adam优化器. 由于多智能体强化学习算法对于网络结构超参数是敏感的,对于QMIX-A* 算法,单智能体网络和混合网络的隐藏层的神经元数量设为16的倍数,为了保持结果的一致性和实验的公平性,IQL-A* 的超参数与QMIX-A* 算法总是相同的.

将所提出的QMIX-A* 算法在路网环境中进行验证,实验分为两个部分:首先,在不同规模的路网上进行不同的工人和任务组合,即每个时间段T内的工人和用户数量是固定的. 为了全方位地展示所提出方法的有效性,这里考虑3种场景:$m<n$, $m=n$, $m>n$. 在第2个实验中,同样针对不同规模的路网,每个时间段T内的工人和用户的数量是随机的,即工人和用户的数量被随机赋值为1到一个预定的最大工人数$m_{\max}$和最大任务数$n_{\max}$区间内的数值,这更加符合现实应用场景. 具体地:对于10×10路网,设置$m_{\max}=10, n_{\max}=10$;对于30×30路网,设置$m_{\max}=15, n_{\max}=15$;对于50×50的路网,设置$m_{\max}=20, n_{\max}=20$. 在不同的空间众包任务场景下训练不同的任务分配算法,记录并比较它们在1 000个相同的测试周期的任务平均完成时间. 更短的时间意味着更好的表现,因为它表示所有用户发布

的任务都能早点完成.

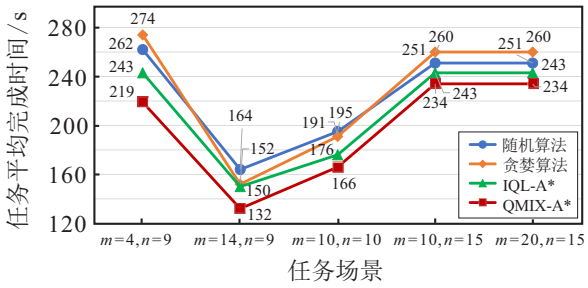
3.2 实验结果

针对固定的任务-工人组合,表1显示:在30×30路网环境中,使用QMIX-A*算法的任务平均完成时间总是最少的;总是选择距离任务起始位置最近的工人的贪婪策略并不总是凑效,尤其当工人数量少于任务数量时. 这是因为按照贪婪原则,任务匹配到的并不总是空闲工人,即可能存在延时匹配的问题.

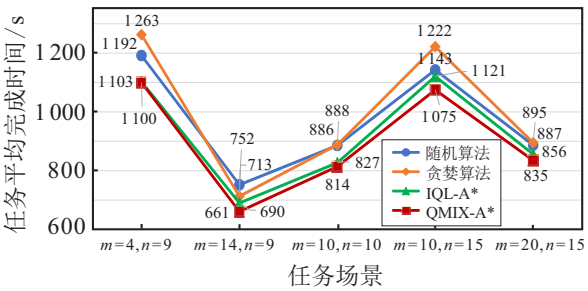
表1 固定的工人-任务组合在30×30路网中的任务平均完成时间 单位: s

方法	$m=4,n=9$	$m=14,n=9$	$m=10,n=10$	$m=10,n=15$	$m=20,n=15$
随机算法	734	460	543	705	545
贪婪算法	779	440	551	759	546
IQL-A*	677	425	512	686	525
QMIX-A*	638	394	460	645	474

为了进一步衡量QMIX-A*算法的扩展性,在计算机设备的性能允许范围内,分别在10 × 10和50 × 50的路网上设置固定工人和任务数量进行测试,图5以折线图的形式记录了固定的任务-工人组合在10×10和50×50路网中的任务平均完成时间. 可以看出,QMIX-A*在不同的任务场景中的表现都优于其他方法.



(a) 不同算法在10×10路网上的分配性能比较



(b) 不同算法在50×50路网上的分配性能比较

图5 固定的工人-任务组合在不同规模路网上的任务分配性能

图6比较了在变化的任务-工人组合的不同规模路网中不同任务分配方法的性能. 当每个时间段T内的工人-用户的数量是随机时,QMIX-A*的表现依然优于其他方法. 图7记录了IQL-A*和QMIX-A*训练50 000个训练周期的任务完成时间的趋势,可以看到:路网规模是10×10时,问题的复杂度不高,IQL-A*尚能表现出较好的性能;但当路网规模增大到30×30和50 × 50时,IQL-A*在任务匹配的动作选择上呈现出随机性,任务完成时间分别在620和1 100附近波动并持续到训练结束,最终也未实现收敛. 随着路网规模的增加,QMIX-A*依然能够保持良好的性能. 随着训练周期的增加,QMIX-A*算法的任务完成时间逐渐下降并大致在30 000个训练周期时收敛到稳定的水平. 这说明QMIX-A*有着更强的学习能力,能够适应更复杂的环境. 此外,智能体数目的增多使得状态空间维度越来越高,也是导致IQL-A*网络权重难以收敛的原因之一. QMIX-A*算法通过全局奖励学习联合动作函数来克服各智能体间存在的通信、决策及执行等问题.

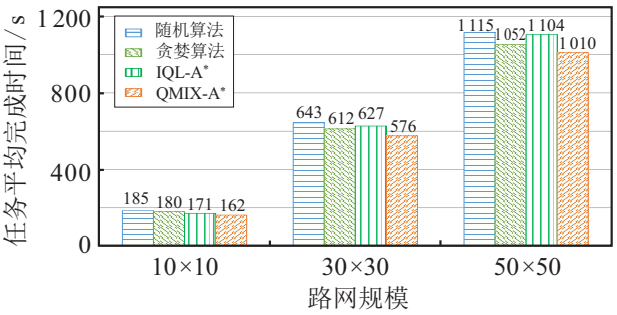


图6 变化的工人-任务组合在不同规模的路网中的任务平均完成时间

4 结 论

针对现有的空间众包任务分配工作忽视路径规划对分配效果的影响,本文提出了面向路网的以任务完成时间最小为目标的实时任务分配与路径优化算法QMIX-A*,可为解决复杂动态供需环境下高效的工人-任务匹配问题提供理论基础与方法储备. 本文的工作可概括为以下几点:

1) 使用多智能体强化学习方法刻画了一个涉及任务匹配和带取送货的路径规划的空间众包场景. 将任务作为智能体,工人集合作为动作空间,基于工人路径规划设计合理的奖励函数,训练任务的匹配选择策略,充分考虑路径和移动成本对工人-任务分

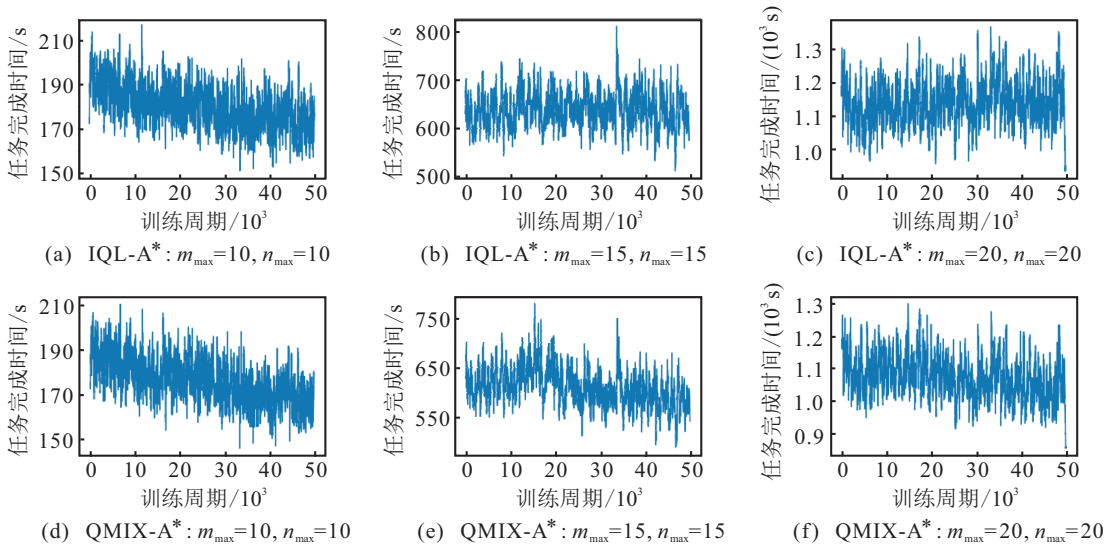


图 7　IQL-A*和QMIX-A*训练50 000个训练周期的任务完成时间

配结果产生的影响.

2) 提出了QMIX多智能体强化学习算法与A*算法融合的QMIX-A*算法. 以QMIX多智能体强化学习算法为基础,将带取送货的路径规划问题抽象为经过必经点最短路径问题,使用A*算法为每个工人规划最短路径进而实现工人-任务间的分配. 为了解决多智能体系统的协同控制问题,QMIX-A*算法采用中心式训练、分布式执行框架,提升了模型的稳定性和优化效果.

3) 进行了大量的数值仿真研究,通过与现有算法进行对比,验证了QMIX-A*算法的有效性、鲁棒性和优越性. 与IQL-A*算法相比,QMIX-A*可获得较好的协同控制效果且能适应更复杂的路网环境.

本文的问题场景中,平台被假设为一开始就知道某时间段内工人和任务的信息. 但是,现实生活中通常是仅在任务或工人到达时才知道其时空信息. 进一步的工作将会针对动态场景进行研究并考虑当前决策对未来收益的影响.

参考文献(References)

[1] Tong Y X, Zhou Z M, Zeng Y X, et al. Spatial crowdsourcing: A survey[J]. The VLDB Journal, 2020, 29(1): 217-250.

[2] 潘庆先, 殷增轩, 董红斌, 等. 基于禁忌搜索的时空众包任务分配算法[J]. 智能系统学报, 2020, 15(6): 1040-1048.
(Pan Q X, Yin Z X, Dong H B, et al. Spatiotemporal crowdsourcing task assignment algorithm based on tabu search[J]. CAAI Transactions on Intelligent Systems, 2020, 15(6): 1040-1048.)

[3] 戴韬, 沈静. 基于众包的外卖配送订单选择研究[J]. 工业工程, 2021, 24(2): 125-133.
(Dai T, Shen J. A research on take-away delivery task selection in crowdsourcing[J]. Industrial Engineering Journal, 2021, 24(2): 125-133.)

[4] 余海燕, 蒋仁莲. 基于众包平台的外卖实时配送订单分配与路径优化研究[J]. 工业工程与管理, 2022, 27(2): 146-152.
(Yu H Y, Jiang R L. Study on the real-time order allocation and routing problem of takeout food distribution on crowdsourcing platform[J]. Industrial Engineering and Management, 2022, 27(2): 146-152.)

[5] To H, Shahabi C, Kazemi L. A server-assigned spatial crowdsourcing framework[J]. ACM Transactions on Spatial Algorithms and Systems, 2015, 1(1): 1-28.

[6] Deng D X, Shahabi C, Demiryurek U. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing[C]. Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Orlando, 2013: 324-333.

[7] 赵杨, 倪志伟, 朱旭辉, 等. 基于改进狮群进化算法的面向空间众包平台的多工作者多任务路径规划方法[J]. 计算机科学, 2021, 48(S2): 30-38.
(Zhao Y, Ni Z W, Zhu X H, et al. Multi-worker and multi-task path planning based on improved lion evolutionary algorithm for spatial crowdsourcing platform[J]. Computer Science, 2021, 48(S2): 30-38.)

[8] 吴腾宇, 陈嘉俊, 蹇洁, 等. O2O模式下的配送车辆实时取送货路径选择问题[J]. 系统工程理论与实践, 2018, 38(11): 2885-2891.
(Wu T Y, Chen J J, Jian J, et al. The online pick-up and delivery vehicle routing problem under O2O delivery[J]. Systems Engineering — Theory & Practice, 2018, 38(11): 2885-2891.)

[9] Safran M, Che D R. Efficient learning-based recommendation algorithms for top-N tasks and top-N workers in large-scale crowdsourcing systems[J]. ACM

Transactions on Information Systems, 2019, 37(1): 1-46.

[10] 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.
(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.)

[11] 王凌, 潘子肖. 基于深度强化学习与迭代贪婪的流水车间调度优化[J]. 控制与决策, 2021, 36(11): 2609-2617.
(Wang L, Pan Z X. Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method[J]. Control and Decision, 2021, 36(11): 2609-2617.)

[12] Li Z N, Yu H, Zhang G H, et al. Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning[J]. Transportation Research — Part C: Emerging Technologies, 2021, 125: 103059.

[13] 王思鹏, 杜昌平, 郑耀. 基于强化学习的扑翼飞行器路径规划算法[J]. 控制与决策, 2022, 37(4): 851-860.
(Wang S P, Du C P, Zheng Y. Local planner for flapping wing micro aerial vehicle based on deep reinforcement learning[J]. Control and Decision, 2022, 37(4): 851-860.)

[14] 李燕君, 蒋华同, 高美惠. 基于强化学习的边缘计算网络资源在线分配方法[J]. 控制与决策, 2022, 37(11): 2880-2886.
(Li Y J, Jiang H T, Gao M H. reinforcement learning-based online resource allocation for edge computing network[J]. Control and Decision, 2022, 37(11): 2880-2886.)

[15] Liu C H, Zhao Y N, Dai Z P, et al. Curiosity-driven energy-efficient worker scheduling in vehicular crowdsourcing: A deep reinforcement learning approach[C]. IEEE 36th International Conference on Data Engineering. Dallas, 2020: 25-36.

[16] Shan C H, Mamoulis N, Cheng R, et al. An end-to-end deep RL framework for task arrangement in crowdsourcing platforms[C]. IEEE 36th International Conference on Data Engineering. Dallas, 2020: 49-60.

[17] Jiang J C, An B, Jiang Y C, et al. Understanding crowdsourcing systems from a multiagent perspective and approach[J]. ACM Transactions on Autonomous and Adaptive Systems, 2018, 13(2): 1-32.

[18] Rashid T, Samvelyan M, de Witt C S, et al. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning[J/OL]. 2018, arXiv: 1803.11485.

[19] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[20] de Lima O, Shah H, Chu T S, et al. Efficient ridesharing dispatch using multi-agent reinforcement learning[J/OL]. 2020, arXiv: 2006.10897.

[21] 田华亭, 李涛, 秦颖. 基于A*改进算法的四向移动机器人路径搜索研究[J]. 控制与决策, 2017, 32(6): 1007-1012.
(Tian H T, Li T, Qin Y. Research of four-way mobile robot path search based on improved A* algorithm[J]. Control and Decision, 2017, 32(6): 1007-1012.)

[22] 郭超, 陈香玲, 郭鹏, 等. 基于时空A*算法的多AGV无冲突路径规划[J]. 计算机系统应用, 2022, 31(4): 360-368.
(Guo C, Chen X L, Guo P, et al. Multi-AGV non-conflict path planning based on space-time A* algorithm[J]. Computer Systems and Applications, 2022, 31(4): 360-368.)

[23] Huber P J. Robust regression: Asymptotics, conjectures and Monte Carlo[J]. The Annals of Statistics, 1973, 1(5): 799-821.

[24] Tampuu A, Matiisen T, Kodelja D, et al. Multiagent cooperation and competition with deep reinforcement learning[J]. PLoS One, 2017, 12(4): e0172395.

作者简介

纪苗苗(1994–), 女, 博士生, 从事众包、机器学习及其应用等研究, E-mail: jimiaomiao@stu.scu.edu.cn;

吴志彬(1982–), 男, 教授, 博士生导师, 从事众包、机器学习及其应用等研究, E-mail: zhibinwu@scu.edu.cn.