张笛. 面向跨业务城市计算的自适应路径规划研究[J]. 智能计算机与应用,2026,16(7):93-99. DOI:10.20169/j.issn.2095-2163.24102804

面向跨业务城市计算的自适应路径规划研究

张 笛

(郑州财税金融职业学院 大数据与人工智能学院, 郑州 450048)

摘 要:随着信息技术的快速发展,城市计算在交通、物流和公共服务等领域的应用日益广泛。然而,异质性与交互需求使得传统路径规划面临新挑战。在多变的城市环境中,路径规划不仅需适应单一业务需求,还需实现不同业务场景之间的有效协同。现有的路径优化算法多集中于特定场景,缺乏跨业务适应能力。为解决这一问题,本文基于深度强化学习提出了一种自适应路径规划方法。该方法通过深度学习自动提取环境特征,并利用强化学习优化路径策略,旨在动态响应交通与物流等多业务场景的需求。实验结果表明,所提方法能够有效适应复杂城市环境下的多发布位置需求,实现高效的路径优化,为未来的城市物流与交通管理提供了新的思路和解决方案。

关键词:强化学习;城市计算;跨业务任务;路径规划

中图分类号:TP18 **文献标志码**:A **文章编号**:2095-2163(2026)07-0093-07

Research on adaptive path planning for cross-business urban computing

ZHANG Di

(College of Big Data and Artificial Intelligence, Zhengzhou Vocational College of Finance and Taxation, Zhengzhou 450048, China)

Abstract: With the rapid development of information technology, urban computing is increasingly used in transportation, logistics, and public services. However, heterogeneity and interaction requirements pose new challenges to traditional path planning. In a changing urban environment, path planning not only needs to adapt to single business needs, but also needs to achieve effective collaboration between different business scenarios. Existing works are mostly focused on specific scenarios and lack cross-business adaptability. To solve this problem, the paper proposes an adaptive path planning method based on deep reinforcement learning. This method automatically extracts environmental features through deep learning and optimizes path strategies using reinforcement learning, aiming to dynamically respond to the needs of multiple business scenarios such as transportation and logistics. Experimental results show that the proposed method can effectively adapt to the needs of multiple release locations in complex urban environments, achieve efficient path optimization. The research provides new ideas and solutions for future city logistics and traffic management.

Key words: reinforcement learning; urban computing; cross-business tasks; path planning

0 引 言

随着信息技术的迅猛发展和智能城市建设的推进,城市计算逐渐应用到社会生活的多个关键领域,包括交通、物流、公共服务和紧急响应等。城市计算技术通过整合城市数据和计算资源,支持城市的高效运营,并且在改善居民生活质量和提高资源利用率方面发挥了显著作用。然而,城市计算中不同业务场景的异质性和交互需求对传统的路径规划技术提出了新的挑战。在多样化和高度动态的城市环境中,路径规划[1]不仅需要适应单一业务场景下的静态或单向需求,还需要跨越业务场景之间的多维度协同,动态适应变化的业务条件,从而实现多任务间的高效资源调度和分配。

现有的城市计算技术在路径规划方面取得了显著的进展,尤其是通过数据驱动的方法,例如基于深度学习和强化学习的路径优化算法[2],能够有效地实现出行需求和资源的动态匹配。城市出行业务中的乘车共享服务[3-5],随着优步、滴滴等平台的推广,使得车辆资源的利用效率和经济效益显著提升[6]。研究者们通过构建共乘模型和时空索引机制,为不同乘客需求之间的动态匹配提供了解决方

作者简介:张 笛(1995—),女,硕士,讲师。主要研究方向:大数据技术,调度优化。Email:se_diz@126.com。

收稿日期:2024-10-28

案,代表性研究包括 Ma 等学者[7]提出的“T-share”系统,该系统通过动态路径规划减少了司机的总行程,提升了整体效率。然而,由于这类系统主要关注单一业务场景,如特定的交通或物流需求,缺乏跨越不同业务场景的适应能力,难以满足复杂的城市计算场景中跨业务需求的动态变化。

同时,城市物流业务的发展也促进了路径规划技术的深入应用。近年来,随着电商和外卖服务的快速增长,物流行业涌现出多种配送模式,如众包配送和基于共享出行的包裹派送,极大地提升了配送效率并节省了成本。Gong 等学者[8]研究了基于社交媒体用户的众包配送方式,使得配送路径得以灵活规划。Chen 等学者[9]提出了通过沿途中转站实现包裹运输的方案,从而降低物流成本。周婷婷等学者[10]研究了恶劣天气下的路径分配问题,提出了一种基于博弈论的任务分配算法 CTMG,虽然 CTMG 能够在多车辆环境中有效分配派送任务,但其计算复杂度较高,且在处理动态变化的路况时可能表现不佳。靳康杰[11]研究了移动机器人的最短路径规划问题,设计出了一种基于 Dijkstra 的路径规划算法 HBSP。而作为一种经典的路径规划算法,具有较高的规划质量,适用于静态环境中的单路径规划,但在多业务路径选择和动态调整方面存在局限性。此外,Lanh 等学者[12]提出了一种基于合作博弈的任务分配算法 CGTM,通过合作博弈来优化任务分配,能够提升多车辆系统的协同效率,但在大规模路网中可能受到计算资源的限制。总地来说,以上这些方法仍然依赖于特定的固定场所或单一业务的任务匹配,对于多业务融合和资源共享的动态需求适应性较差。在多业务交互需求日益增加的背景下,现有方法的局限性使得传统路径规划在应对复杂多源场景时难以满足效率和实时性要求。

为了解决上述不足,本文基于深度强化学习技术,提出了一种适用于跨业务城市计算场景的自适应路径规划方法。该方法通过深度学习实现对环境特征的自动提取与建模,同时利用强化学习自适应地优化派送路径策略,旨在应对城市环境中的多变因素和跨业务需求。具体而言,本文的方法不仅能够动态响应交通和物流等多业务场景间的任务需求,还能在复杂多变的城市环境中实现路径的高效优化,从而满足多源动态数据下的路径规划要求。

1 场景描述

为更直观地说明跨业务路径规划问题,本文以物流业务与拼车业务为例,构建了一个跨业务路径规划场景(如图 1 所示)。在图 1 中,出租车 1 在始发地 A 接收了一个需要运输至中转站 B 的包裹。在运输过程中,平台为其分配了一位行程相似的乘客 1。出租车 1 首先将包裹送至中转站点 B 并放下,随后继续为乘客 1 提供服务。与此同时,另一辆前往目的地 C 的出租车 2 出现,平台将包裹重新分配给出租车 2。出租车 2 随后从中转站 B 出发,在服务乘客 2 的同时,将包裹运输至目的地 C。上述场景中,出租车既承担了乘客运输任务,又执行了包裹配送任务,体现了跨业务任务的协同特点。本文研究的重点在于如何优化车辆路径规划,以更高效地完成跨业务任务分配,从而提升车辆利用率和平台整体收益。由于包裹派送和打车业务具有时空相似性,因此本文把包裹分配和打车的路径规划进行了统一处理,并未设置单独的包裹管理模块。



图 1 跨业务路径规划示例图

Fig. 1 A toy example of cross-business route planning

1.1 路网建模

本文将路网建模为一个无向联通图 $G=(V,E)$,其中包含节点集(V)和边集(E),并定义了相应的距离矩阵($\boldsymbol{D}$)和车流矩阵($\boldsymbol{L}$),详细说明如下:$V=\{v_1,v_2,\cdots,v_n\}$ 表示节点集,每个节点 v_i 代表道路网络中的一个交叉口;$E=\{e_1,e_2,\cdots,e_n\}$ 表示边集,每条边 e_i 代表连接 2 个节点的路段。

距离矩阵 $\boldsymbol{D}$ 表示如下:

$$\boldsymbol{D}=\begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix} \tag{1}$$

其中,d_{ij} 表示 v_i 和 v_j 之间的距离。车流矩阵 $\boldsymbol{L}$ 表示如下:

$$\boldsymbol{L}=\begin{bmatrix} l_{11} & \cdots & l_{1n} \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \tag{2}$$

其中，l_{ij} 表示从 v_i 到 v_j 的车辆流量。

为表征城市路网中速度与密度的关系，本节使用了 Aoki[13] 提出的速度-密度指数模型。该模型能够体现不同车型对速度的影响。具体地，车辆速度 s 表示如下：

$$s = s_0 \omega_{ij}^{\sigma} \tag{3}$$

其中，s_0 表示车辆的自由流速（如 25 km/h）；σ 为缩放因子（如-0.7）；ω_{ij} 表示按道路宽度计算的车辆密度，定义公式为：

$$\omega_{ij} = l_{ij} / d_{ij} \tag{4}$$

用 c_{ij}^t 表示从节点 v_i 到 v_j 的时间成本。基于式（3），c_{ij}^t 可以表示为：

$$c_{ij}^t = \frac{d_{ij}}{s} \tag{5}$$

由于道路长度 d_{ij} 和自由流速 s_0 固定，因此 c_{ij}^t 随着密度 ω_{ij} 的增加而增大，即时间成本增加。

1.2　车流量建模

本文利用时间序列方法对车流的变化趋势进行预测，旨在建立当前时间片车辆数量与前几个时间片车流量之间的线性关系。鉴于车流在短期内的相关性特征，本节采用自回归综合移动平均模型[14]（Auto Regressive Integrated Moving Average，ARIMA）来评估时间序列的动态性与连续性，从而刻画历史与当前数据间的关系。

设差分算子为 ∇，则一阶差分 $\nabla f_{ij}(x)$ 和 k 阶差分 $\nabla^k f_{ij}(x)$ 表示如下：

$$\nabla f_{ij}(x) = f_{ij}(x) - f_{ij}(x-1) \tag{6}$$

$$\nabla^k f_{ij}(x) = \nabla^{k-1} f_{ij}(x) - \nabla^{k-1} f_{ij}(x-1) \tag{7}$$

引入延迟算子 ϑ，使得：

$$\lambda_{ij}(x-p) = \vartheta^p \lambda_{ij}(x) \tag{8}$$

则有：

$$\nabla^k = (1-\vartheta)^k \tag{9}$$

假设 k 阶差分后的序列 $\{f_{ij}\}$ 是平稳的，对应的 ARIMA 模型可表达为：

$$\rho(\vartheta) \nabla^k f_{ij}(x) = \xi(\vartheta) \tau(x) \tag{10}$$

其中，$\tau(x)$ 表示通用的白噪声序列，满足均值 $\mathbb{E}[\tau(x)] = 0$、方差 $\mathrm{Var}[\tau(x)] = \delta^2$，且每个序列点均服从正态分布。在式（10）中，自回归系数多项式 $\rho(\vartheta)$ 和移动平均平滑系数多项式 $\xi(\vartheta)$ 定义为：

$$\rho(\vartheta) = 1 - \sum_{p=1}^{k_1} \rho_p \vartheta^p \tag{11}$$

$$\xi(\vartheta) = 1 - \sum_{p=1}^{k_2} \xi_p \vartheta^p \tag{12}$$

其中，正整数 k_1 和 k_2 分别表示自回归和移动平均模型的阶数。

2　问题描述

2.1　MDP 建模

依据路网模型的特性，可以将基于寻路策略的环境探索[15-16] 建模为一个马尔可夫决策过程（Markov Decision Process，MDP），其目标是最大化累积总奖励。MDP 通常表示为一个五元组 $M = \{S, A, R(s' \mid s,a), P(s' \mid s,a), \gamma\}$。具体地，$S$ 表示状态空间；A 表示动作空间；$R(s' \mid s,a)$ 表示在状态 s 下执行动作 a 转移到状态 s' 所获得的即时奖励；$P(s' \mid s,a)$ 表示在状态 s 下执行动作 a 转移到状态 s' 的概率；$\gamma \in [0,1]$ 为折扣因子。

考虑到每个路口与唯一坐标的对应关系，本文将智能体（即车辆）当前位置的观察记作状态 s_t。对于大小为 $N \times M$ 的网格图，其状态空间可以表示为：

$$s_t = \{g_{(x,y)}, l_{(x,y)}, l_{(x-1,y)}, l_{(x+1,y)}, l_{(x,y-1)}, l_{(x,y+1)}, \tau_{(x,y)}, \tau_{(x-1,y)}, \tau_{(x+1,y)}, \tau_{(x,y-1)}, \tau_{(x,y+1)}\} \tag{13}$$

其中，$g_{(x,y)}$ 表示当前智能体所在网格，$x \in [0, N]$，$y \in [0, M]$；$l_{(x,y)}$ 表示网格 $g_{(x,y)}$ 未来 T 个时间片内的平均车流量；$\tau_{(x,y)}$ 表示网格 $g_{(x,y)}$ 中跨业务任务的数量（若存在 n 类业务，则 τ 为一个 $1 \times n$ 的向量）。

在每个时间片 t 中，智能体在接收环境奖励 r_t 之前，会依据策略 $\pi(a_t \mid s_t)$ 选择一个动作 a_t。基于当前位置的观察，智能体能够识别一组可达节点，其动作空间可以表示为：

$$A = \{\text{中}, \text{上}, \text{下}, \text{左}, \text{右}\} \triangleq \{0,1,2,3,4\} \tag{14}$$

每个元素代表网格内不同方向的移动，其中“上”表示从当前网格移动到相邻的 $(x, y+1)$ 网格。然而，当智能体处于地图边缘时，其有效动作空间实际上是 A 的一个子集。值得注意的是，若智能体选择的方向超出了当前地图所定义的区域，该动作将被视为无效，智能体需重新选择剩余有效方向之一。

智能体的目标在于找到最优的行为策略，以最大化每个状态 s_t 的长期折扣奖励，即最大化以下公式：

$$r_t^{\gamma} = \sum_{i=t}^{T} \gamma^{i-t} r_t \tag{15}$$

其中，T 为终止时间片；折扣因子 γ 用于平衡即时奖励与未来奖励的重要性。在本文中，终止时间片被设置为智能体完成路径规划（即从当前网格移

动到目标网格)所需时间片的数量。这一设定主要依据道路的特性(如路段长度和车辆速度)以及实际应用场景的需求。时间片的选取需平衡实时性与计算效率。较短的时间片能够捕捉更多的环境动态,但会增加计算复杂度;较长的时间片则降低了对环境细节的敏感性,但适合于变化较慢的场景。

每当智能体执行一个动作后,就会从环境中获得反馈。假设智能体在执行选定动作后成功从节点 v_i 移动到节点 v_j,则即时奖励函数可表示为:

$$r_t = \begin{cases} -c_{ij}^t(t) + \kappa(\eta(t) - \eta(t-1)), & \text{其他} \\ -c_{ij}^t(t) + \kappa(\eta(t) - \eta(t-1)) + k, s_t = s_{终止} \end{cases} \tag{16}$$

其中,$\eta(t)$ 表示时间片 t 时从当前位置到目的地的距离,κ 为权重系数。式(16)为分段函数,主要考虑 $s_t = s_{终止}$ 和其他情况。在其他情况下,第 1 项表示智能体通过当前路段所耗费的时间,是决定当前行动所获奖励的关键因素。第 2 项则起到动机激励的作用,旨在促使智能体向目标前进,避免在训练过程中出现无效的徘徊行为。通过灵活设定这些参数的取值,奖励函数能够适应不同业务场景,从而支持跨业务路径规划的多样化目标。

在训练的早期阶段,智能体可能因距离目标较远而难以做出正确判断,因此式(16)中的第 2 项可以帮助智能体避免在相似路况下选择错误的方向。值得注意的是,在本文中,κ 应该设置得相对较小,因为时间消耗是路径规划中需重点考虑的因素:即在节点间的时间消耗越小,包裹完成的概率越高。在 $s_t = s_{终止}$ 的情况下,式(16)中的“$+k$”表示当智能体到达目的地时,将获得额外的正奖励。

2.2 问题定义

在车辆路径规划过程中,针对同一起点和终点,路网中可能存在多条可行的路径。本文旨在寻找从起始节点到目标节点所需时间最短的最优节点序列(即策略)。设 $\{a_1, a_2, \cdots, a_T\}$ 为智能体的完整动作序列,则跨业务路径规划问题可以定义为:

$$\min_{a_t} \pi \left[\sum_{t=1}^{T} r_t \right] \tag{17}$$

其中,$\pi[\cdot]$ 表示策略集,且 $a_t \in A$ 表示动作空间中的一个元素。

需要强调的是,尽管本节定义的车辆路径规划问题的优化目标是最小化路径时间成本,但其核心目的仍然是最大化平台的整体收益。一条时间成本较低的站点路径有助于提高跨业务请求完成率,从而提升平台的总收益。

3 自适应路径规划算法

在城市计算的跨业务路径规划场景中,由于车流矩阵 $\boldsymbol{L}$ 的动态变化,车辆在节点间的转移时间通常难以准确估计。为应对这一挑战,本节提出了一种基于深度强化学习的自适应路径规划(Adaptive Route Planning,ARP)算法。该算法通过捕捉实时环境中的车流变化,为每台车辆自适应地规划出一条最佳行驶路径。

ARP 算法的学习框架如图 2 所示。首先,智能体观察环境状态并选择相应的动作来执行。其次,环境根据智能体的动作反馈奖励值,并进入下一个状态。这些元素在时间步 t 中构成一个经验元组 $\varphi_t = (s_t, a_t, r_t, s_{t+1})$。最后,智能体从经验回放缓冲区随机采样一批记忆经验,并计算损失函数如下:

$$L(\theta_i) = \mathbb{E}_{\varphi_t \sim \Pi}[(y_i - Q(s_t, a_t; \theta_i))^2] \tag{18}$$

其中,

$$y_i = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a_{t+1}; \theta^-) \tag{19}$$

其中,θ^- 是旧的目标网络参数,在一段时间后会被评估网络参数 θ_{i-1} 替换,从而实现迭代更新。

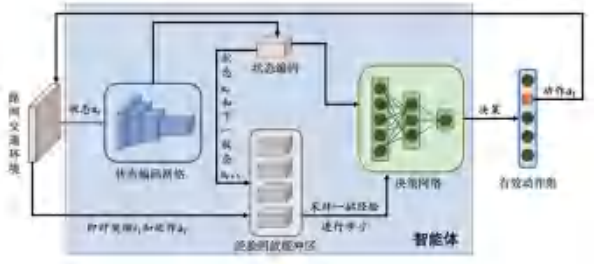


图 2 ARP 算法的学习框架

Fig. 2 Learning framework of ARP algorithm

综上,本节提出了 ARP 算法,算法代码具体如下。

算法 1 ARP 算法

1. 初始化经验回放缓冲区 Π;
2. 用网络参数 θ 初始化 Q 网络;
3. 用 $\theta^- = \theta$ 初始化目标 Q 网络;
4. for 迭代轮次 episode←1 do
5. 初始化环境,观测初始位置 s_1;
6. for 时间 $t = 1, 2, \cdots, T$ do
7. 以 $1 - \varepsilon$ 的概率贪婪或 ε 的概率按学习到的策略选择动作 a_t;
8. 往 a_t 方向移动,从环境中获得奖励 r_t,并观测下一位置 s_{t+1};
9. 收集决策经验并将经验元组 φ_t 存储到 Π 中;

10. 随机从 Π 中采样一批经验(s_i,a_i,r_i,s_{i+1});
11. if 时间 $i+1$ 为终止状态 then
12. 利用式(19)计算 y_i;
13. else
14. 令 $y_i = r_t$;
15. end if
16. 执行梯度下降并利用式(18)更新 Q 网络;
17. 更新当前环境,并每 k 步令 $\theta^- = \theta$;
18. end for
19. end for

4　实验评估

为评估本文提出的基于强化学习的自适应路径规划(ARP)算法的效率与性能,将其与几种同领域的先进算法进行对比分析。

4.1　实验设置

(1)数据集:本文使用的数据集为由滴滴公司和 NYCTaxi 发布的成都和纽约真实出租车数据集[17]。成都数据集包含 209 423 个订单、36 630 个节点和 50 786 条边;纽约数据集包含 10 906 858 个订单、264 346 个节点和 366 923 条边。每个订单均与一个唯一的 ID、订单生成时间、完成时间以及上下车地点相关联。为了支持不同的业务需求,本文将数据集中一部分订单随机设置为乘客任务,另一部分设置为包裹派送任务。这种任务划分方式基于其相似的起点-终点特征和时空规律,在城市计算领域是一种常见的处理方法。

(2)对比方法:本文使用的对比算法包括:基于博弈论的任务分配算法(CTMG)[10]、基于 Dijkstra 的路径规划算法(HBSP)[11]以及基于合作博弈的任务分配算法(CGTM)[12]。

(3)实验指标:本文采用 3 个主要评价指标,分别为总收益[18]、响应时间[19]及任务完成率[20]。其中,总收益为所有车辆与跨业务请求匹配的收益之和。具体来说,每个车辆-跨业务请求匹配的收益为请求价格减去车辆为了完成该请求所产生的移动花销,计算公式为:

$$S_i = \mathrm{Pr}_i - \mathrm{Mo}_i \tag{20}$$

其中,S_i 表示第 i 次收益;Pr_i 表示第 i 次请求价格;Mo_i 表示第 i 次移动花销。

接着,将所有车辆和请求匹配的收益相加,以获得总收益,计算公式为:

$$\mathrm{SBe}_T = \sum_{j=1}^{n} \mathrm{Be}_j \tag{21}$$

其中,SBe_T 表示总收益;Be_j 表示第 j 次收益;n 表示成功匹配的车辆和请求对的总数。

(4)实验环境:实验中的所有算法均采用 Python 实现,神经网络框架为 TensorFlow 1.15.0。在配置为 Intel i7-9700K @ 3.6 GHz CPU 和 16 GB RAM 的计算机上进行实验。

4.2　实验结果与分析

(1)成都数据集。图 3~图 5 展示了在站点数量(即跨业务任务发布位置点)从 50~250 变化过程中,4 种算法在总收益、响应时间和任务完成率三个指标上的性能比较。从图 3 和图 5 可见,采用 HBSP 算法的 HBSP+CTMG 和 HBSP+CGTM 在总收益和任务完成率上表现出先增加后下降的趋势。这一现象可归因于 HBSP 算法为基于 Dijkstra 的站点规划方法,随着站点数量的增加,路径规划的精确度有所提高。然而,站点数量过多可能使得规划出的路径跳数增加,进而导致大部分任务未能及时完成,造成总收益和任务完成率下降。此外,基于 Dijkstra 的算法在站点数量增加时的时间开销也随之增加,这一点在图 3 中得到了验证。

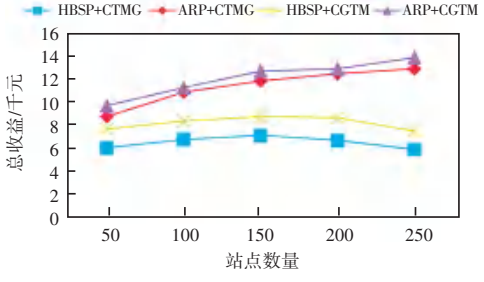


图 3　总收益与站点数量关系

Fig. 3　Relationship between total revenue and number of stations

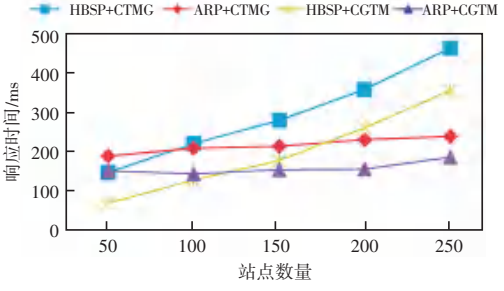


图 4　响应时间与站点数量关系

Fig. 4　Relationship between response time and number of stations

与此不同,采用 ARP 算法的 ARP+CTMG 和 ARP+CGTM 在总收益和任务完成率方面持续上升,没有出现拐点。这是因为 ARP 算法基于深度强化学习,能够根据实时环境进行自适应路径规划,从而有效提高了路径的规划质量。此外,ARP 考虑了任务过期的因素,因此随着站点数量的增加,规划出的

路径不仅精确,还避免了因跳数过多导致的任务过期问题。由图 4 可知,基于 ARP 的算法未出现“站点越多、响应时间越高”的趋势。这是因为深度强化学习的决策由一个经过良好拟合的状态-动作值函数[21]的驱动,其表现不受站点数量的直接影响。

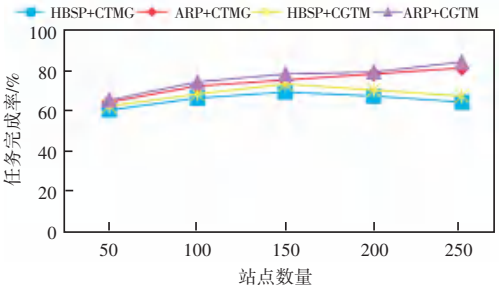


图 5 任务完成率与站点数量关系

Fig. 5 Relationship between task completion rate and number of stations

(2)纽约数据集。图 6~图 8 展示了在纽约数据集上,随着站点数量从 500 增加到 2 500 时,4 种算法在总收益、响应时间和任务完成率三个指标上的性能对比。从图 6 和图 8 可以看出,采用 HBSP 算法的 HBSP+CTMG 和 HBSP+CGTM 在总收益和任务完成率方面表现出先上升后下降的趋势。相对而言,采用 ARP 算法的 ARP+CTMG 和 ARP+CGTM 在这 2 个指标上则持续增长,没有显著的拐点。此外,图 7 显示,随着站点数量的增加,HBSP 算法的 HBSP+CTMG 和 HBSP+CGTM 的响应时间不断上升,而 ARP 算法的 ARP+CTMG 和 ARP+CGTM 的响应时间则保持相对稳定。这些现象的产生原因与图 3~图 5 中的分析相似,因此不再详细讨论。

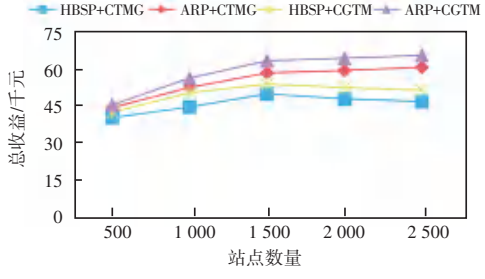


图 6 总收益与站点数量关系

Fig. 6 Relationship between total revenue and number of stations

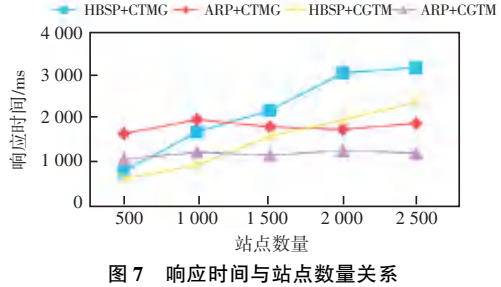


图 7 响应时间与站点数量关系

Fig. 7 Relationship between response time and number of stations

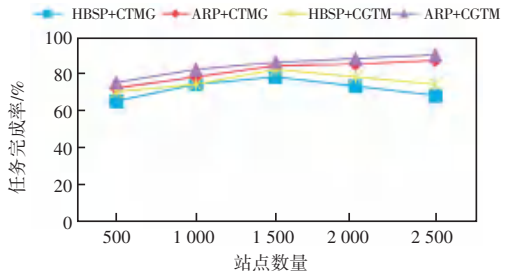


图 8 任务完成率与站点数量关系

Fig. 8 Relationship between task completion rate and number of stations

5 结束语

本文针对跨业务城市计算中的路径规划问题进行了深入分析,提出了一种 ARP 算法。通过在成都和纽约数据集上进行的实验结果表明,ARP 算法在总收益、响应时间和任务完成率等指标上整体优于传统算法。特别地,ARP 算法有效提升了路径规划准确性,避免了因跨业务任务发布位置过多导致的任务过期问题,同时保持了响应时间的稳定性。

参考文献

[1] 何杏宇, 王威, 杨桂松, 等. 面向实时环境的车机协同包裹配送路径规划[J]. 智能计算机与应用, 2024, 14(7): 10-19.

[2] Qu S, Hu G Q. Scalable learning for multiagent route planning: Adapting to diverse task scales [J]. IEEE Transactions on Artificial Intelligence, 2024,5(10): 4996-5011.

[3] Cheng P, Xin H, Chen L. Utility-Aware Ridesharing on Road Networks [C]//Proceedings of the 2017 ACM International Conference on Management of Data. New York: ACM, 2017: 1197-1210.

[4] 胡忠恺,袁鹏程. 共享出行问题及求解算法研究综述[J]. 智能计算机与应用, 2021, 11(4): 164-169.

[5] Luo Y C, Fu S J, Jia X H, et al. P^2 Ride: Practical and privacy-preserving ride - atching scheme for ride sharing [J]. IEEE Transactions on Intelligent Transportation Systems, 2023,24(3): 3584-3593.

[6] Jin Q Q, Li B Y, Cheng Y R, et al. Real-time multi-platform route planning in ridesharing [J]. Expert Systems with Applications, 2024, 255: 124819.

[7] Ma Shuo, Zheng Yu, Wolfson O. T-share: A large-scale dynamic taxi ridesharing service [C]//Proceedings of the International Conference on Data Engineering. Piscataway, NJ: IEEE, 2013: 410-421.

[8] Gong W, Zhang B X, Li C. Task assignment in mobile crowdsensing: Present and future directions[J]. IEEE Network, 2018, 32(4): 100-107.

[9] Chen L, Zhong Q L, Xiao X K, et al. Price and time aware dynamic ridesharing [C]//Proceedings of the International Conference on Data Engineering. Piscataway, NJ: IEEE, 2018: 1061-1072.

[10] 周婷婷, 隋东, 杨振宇, 等. 基于博弈的恶劣天气下航班路径分配研究[J]. 航空计算技术, 2023, 53(6): 30-34.

[11]靳康杰. 基于 Dijkstra 算法的移动机器人最短路径规划设计研究[J]. 信息与电脑(理论版), 2024, 36(8): 47-49.
[12] Lanh V, Manh D, Quang P. Game theory - based optimal cooperative path planning for multiple UAVs[J]. IEEE Access, 2022, 10: 108034-108045.
[13] Aoki T. The simulation system of passenger flow[R]. Tokyo: RTRI, 1994:35.
[14]张腾飞, 袁鹏程. 基于 ARIMA 的短时交通量预测模型[J]. 智能计算机与应用, 2020, 10(7): 273-278.
[15]陈玥. 基于深度强化学习的图书分拣车的路径规划[J]. 现代电子技术, 2024, 47(19): 161-168.
[16]夏雨奇, 黄炎焱, 陈恰. 基于深度 Q 网络的无人车侦察路径规划[J]. 系统工程与电子技术, 2024, 46(9): 3070-3081.
[17]Li Y F, Wu Q S, Huang X, et al. Efficient adaptive matching for real - time city express delivery [J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 5767-5779.
[18]Liu Z, Xiao G Q, Zhou X, et al. Cross online assignment of hybrid task in spatial crowdsourcing[C]//Proceedings of the 40th IEEE International Conference on Data Engineering. Piscataway, NJ: IEEE,2024: 317-329.
[19]Wang Z W, Zhao Y, Chen X H, et al. Task assignment with worker churn prediction in spatial crowdsourcing [C]// Proceedings of the International Conference on Information and Knowledge Management. Piscataway, NJ: IEEE, 2021: 2070-2079.
[20]Zheng T T, Chen J X, Huang Y C . Task assignment in business processes based on completion rate evaluation[C]// Proceedings of the International Conference on Computer, Network, Communication and Information Systems. Amsterdam, the Netherlands: Atlantis Press, 2019: 326-333.
[21]杨明川,朱敬华,李元婧,等. 一种考虑隐私保护的深度强化学习任务分配模型[J]. 计算机研究与发展, 2023, 60(11): 2650-2659.