面向多楼层取送货问题的深度强化学习稀疏跨层路径优化方法

孟 运，李 超†，孙 俊

(江南大学 人工智能与计算机学院，江苏 无锡 214122)

摘 要: 基于自动化运输技术的取送货问题在多楼层环境中面临严峻挑战, 其主要原因在于楼层间电梯数量有限形成的子区域间稀疏连接拓扑, 使得现有基于深度强化学习 (Deep Reinforcement Learning, DRL) 的路径优化方法难以规划出跨楼层的优质路径. 为此, 本文提出一种基于 DRL 的稀疏跨层路径优化 (DRL-based Sparse Cross-Floor Optimization, DRL-SCFO) 方法. DRL-SCFO 的编码器同时提取楼层内与楼层间的空间特征, 并采用自适应策略融合二者, 从而有效地建模复杂的跨楼层拓扑关系. 解码器将每个决策步骤分为两阶段: 首先进行全局路径规划确定潜在的指引节点, 然后以该节点为导向选择局部路径点. 这种结合了全局与局部视角的机制, 增强了模型的空间感知能力和决策准确性. 大量实验表明, DRL-SCFO 在解决多楼层取送货问题上显著优于现有的基于 DRL 的路径优化方法, 并且能够适用于不同的任务拓扑结构与问题规模.

关键词: 深度强化学习；取送货问题；注意力机制；双阶段解码器

中图分类号: TP181 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2026.0120

引用格式: 孟运, 李超, 孙俊. 面向多楼层取送货问题的深度强化学习稀疏跨层路径优化方法 [J]. 控制与决策, xxxx, x(x): xxxx-xxxx.

DRL-SCFO: deep reinforcement learning-based sparse cross-floor path optimization for multi-floor pickup and delivery problems

MENG Yun，*LI Chao*†，*SUN Jun*

(School of Artificial Intelligence and Computer Science，Jiangnan University，Wuxi 214122，Jiangsu，China)

Abstract: Pickup and delivery problems in multi-floor environments pose significant challenges for automated transportation systems, mainly due to the sparse inter-floor connectivity caused by the limited number of elevators. This results in loosely connected subregions, which makes it difficult for existing path optimization methods to generate high-quality cross-floor routes. Although deep reinforcement learning (DRL) has shown strong potential in vehicle routing problems, most DRL-based approaches struggle to effectively model such sparse cross-floor topologies. To address this issue, this paper proposes a DRL-based sparse cross-floor path optimization (DRL-SCFO) method. In DRL-SCFO, the encoder jointly extracts spatial features within individual floors and across different floors, and integrates them through an adaptive fusion strategy to capture complex cross-floor topological structures. The decoder decomposes each decision step into two stages: a global planning stage to determine a potential target, followed by a local execution stage to select a feasible path node guided by this target. Experimental results demonstrate that DRL-SCFO achieves superior solution quality compared with existing DRL-based routing optimization methods on multi-floor pickup and delivery problems, and shows good adaptability across different task topologies and problem scales.

Keywords: deep reinforcement learning；pickup and delivery problem；attention mechanism；dual-stage decoder

0 引 言

取送货问题 (Pickup and Delivery Problem, PDP) 是经典车辆路径问题 (Vehicle Routing Problem, VRP) 的一个重要拓展, 旨在为一组涉及货物或乘客的运输需求规划出货运或客运的最优行驶路径[1]. 随着物联网技术的快速发展, 运输机器人等智能设备已能

收稿日期: 2026-02-03；**录用日期:** 2026-08-19.

基金项目: 国家自然科学基金 (62503198, 62272202), 江苏省自然科学基金 (BK20221068).

责任编委: 李勇建.

†通信作者. E-mail: chaoli@jiangnan.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.

够自主操控电梯, 这使得在多楼层环境中高效求解 PDP 的需求日益迫切. 图 1 展示了一个多楼层仓储环境中的实际 PDP 场景: 自动运输设备不仅需要完成同层配送 (如 P6/F2→D6/F2), 还需执行跨楼层配送任务 (如 P1/F3 → D1/F1). 可以看到, 图 1 中不同楼层间仅通过数量有限的电梯连接, 这形成了一种多个子区域间只具有稀疏连接通道的复杂拓扑结构. 类似的场景还包括酒店机器人送货[2]、多层仓库自动搬运[3] 等. 在本文中, 我们将具有此类特殊拓扑结构的 PDP 定义为多楼层 PDP (Multi-Floor PDP, MF-PDP). 与传统的在单一连续平面上的 PDP 不同, MF-PDP 中子区域间的稀疏连通特性对主流路径优化方法带来了重大挑战.

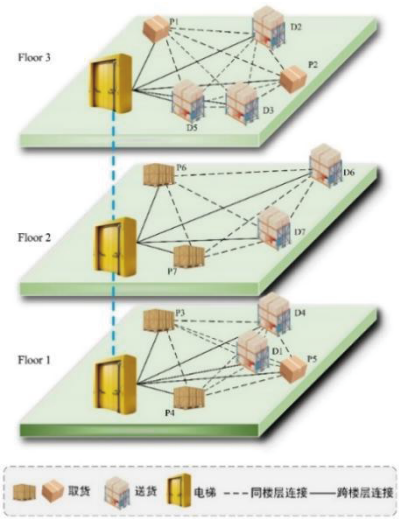

图 1 多楼层取送货问题

传统的 PDP 求解方法涵盖精确算法、近似算法及启发式算法等[4]. 其中, 精确算法通常仅适用于小规模问题, 近似与启发式算法则以牺牲最优性为代价来提升求解效率. 然而, 这些方法普遍缺乏跨问题实例的泛化能力, 每次求解均需重新执行完整的计算流程, 基本不具备实时求解高质量解的能力. 为克服效率与泛化性瓶颈, 近年来, 基于深度强化学习 (Deep Reinforcement Learning, DRL) 的路径优化方法[5-7] 通过学习捕获问题结构的参数化策略, 在解决组合优化问题方面显现出很大潜力. 一旦 DRL 模型被训练完成, 即可在相似实例上以极低的资源消耗生成高质量的解, 这在工程问题上具有极高的应用价值.

然而, 大多数现有的基于 DRL 的方法采用单一的网络结构对所有节点进行编码, 未能有效刻画 MF-PDP 中节点因所属楼层不同而产生的空间特异性. 具体而言, 这类方法通常使用同一个编码模块来对所有节点对之间的交互进行建模, 而不考虑 MF-PDP 中由于楼层间电梯数量有限而导致的楼层间连接稀疏等实际情况, 这可能使得模型会在“所有节点均等同可达”的错误前提下进行训练, 从而致使其难以有效表示由稀疏连接的子区域所构成的复杂层次化拓扑结构.

此外, 当前多数求解 PDP 的 DRL 方法均采用基于编码器-解码器的自回归架构[6-11], 其解码器在每一步决策时主要依赖上一个时间步所选节点的局部 (一跳) 邻域信息, 这在解决传统的单层平面 PDP 上是有效的. 然而, 在 MF-PDP 中, 运输设备必须以电梯为媒介到达其他楼层的取送货节点. 因此, 由于 MF-PDP 特殊的拓扑性质, 解码器的决策范围常被限制在当前楼层内, 这严重制约了其全局探索的能力, 从而影响了最终解的质量.

为解决上述问题, 本文提出了一种基于 DRL 的 MF-PDP 路径优化 (DRL-based Sparse Cross-Floor Optimization, DRL-SCFO) 方法. 该方法包括一个双视域空间注意力编码器 (Dual-Horizon Spatial Attention encoder, DSAenc) 以及一个“潜在规划-实际执行”双阶段解码器 (Potential-to-Real Dual-Stage decoder, PDSdec). 在多种规模的 MF-PDP 实例及退化到单层 PDP 上的实验结果表明, DRL-SCFO 均优于多种主流 DRL 路径优化方法, 并展现出良好的泛化能力.

本文的主要贡献如下:

(1) 对具有子区域间连接稀疏特征的 PDP 新变体——多楼层取送货问题 (MF-PDP) 进行了形式化定义和数学建模.

(2) DSAenc 使用不同的掩码分别提取同楼层与跨楼层的空间信息, 并引入自适应融合策略进行不同区域特征的有效整合与权衡.

(3) PDSdec 将每一次决策划分为全局规划与局部节点选择两个阶段, 通过信息传递与参数共享实现协同决策, 以应对问题的特殊拓扑结构导致的决策范围受限问题.

1 研究现状分析

强化学习 (Reinforcement Learning, RL) 通过与环境交互学习策略, 无需依赖人工设计的启发式规则或标注的最优解, 在 VRP 中表现出较好的应用潜力[12]. 通常, 基于 RL 的 VRP 求解方法可分为两类: 构建式方法 (construction methods) 和改进式方法 (improvement methods)[13]. 主流的改进式方法[14-16] 以对初始解的迭代优化为核心, 从而提升解的质量. 然而, 这类方法的迭代过程通常假设解路径的长度是

固定的. 这一假设限制了其在求解 MF-PDP 时的适用性. 在该问题中, 由于电梯可能被重复使用, 解路径的长度不再保持固定, 而是随决策过程动态变化. 鉴于上述局限性, 本文提出的 DRL-SCFO 采用构建式方法进行求解, 以避免对路径长度的预先假设.

构建式方法通过在每个决策步骤中顺序选择下一个节点来逐步构建完整路径. 这一方法最早由指针网络 (Pointer Networks, Ptr-Net)[17] 引入, 实现了对可变长度输出序列的端到端学习. 早期的 Ptr-Net 主要采用监督学习进行训练. 随后, Bello 等人[12] 将强化学习引入 Ptr-Net, 取得了关键突破, 从而避免了对昂贵最优解标签的依赖.

在此基础上, Kool 等人[5] 提出了注意力模型 (Attention Model, AM), 其采用基于 Transformer 的编码器-解码器架构取代了 Ptr-Net 的主干网络, 在多个组合优化基准问题上取得了显著性能提升, 并成为后续研究的重要基础. 例如, 多重最优策略优化 (Policy Optimization with Multiple Optima, POMO)[18] 利用组合优化解的内在对称性, 通过从多个不同起点并行进行策略推演 (rollouts), 显著提升了训练过程的稳定性并有效缓解了局部最优问题; 多解码器注意力模型 (Multi-Decoder Attention Model, MDAM)[19] 则通过引入多解码器结构以增强路径解的多样性. 近期, Zhou 等人提出了多任务车辆路径求解器 (Multi-Task Vehicle Routing Solver with Mixture-of-Experts, MVMoE)[7]. MVMoE 采用混合专家架构, 在多任务学习与跨路径任务中展现出良好的泛化能力. 此外, Son 等人[11] 提出的 RRNCO 方法通过自适应节点嵌入与神经自适应偏置联合编码边特征, 增强了对节点间空间关系的建模能力. 王万良等[20] 面向多配送中心车辆路径问题, 构建了基于注意力机制与掩码机制的多智能体策略网络, 并结合局部搜索提升解的质量; 周梦等[21] 针对带二维装箱约束的车辆路径问题, 提出知识驱动的强化学习算法, 实现了路径与装载约束的协同求解. 然而, 在 MF-PDP 中, 上述方法在编码阶段未显式区分节点类型, 而由于跨楼层移动的成本通常高于同楼层内的移动, 因此存在一定局限性.

尽管相关研究仍较为有限, 已有少量工作尝试通过建模节点间关系来处理 PDP 中的复杂结构约束. 其中, 异构注意力模型 (Heterogeneous Attentions Model, Heter-AM)[6] 对注意力机制进行了改进, 以显式区分取货与送货节点, 从而更有效地编码“先取后送”的优先级约束. 从理论上看, 可以通过为不同楼层、电梯或楼层-电梯组合设计独立的注意力类型, 将 Heter-AM 扩展到 MF-PDP; 然而, 这种设计会导致注意力类型数量随楼层规模快速增长, 从而在大规模问题实例中降低训练与推理效率.

此外, AM 及其多数变体采用的解码器架构较大程度依赖于单跳 (one-hop) 邻域信息[5]. 当应用于 MF-PDP 时, 这类解码策略往往倾向于在当前楼层内选择局部可达节点进行移动, 从而忽略了可能涉及多次电梯转换的全局最优路径, 在一定程度上影响了整体解的质量.

2　MF-PDP 的形式化定义

本节将多楼层取送货问题 (MF-PDP) 形式化定义为一个图优化问题, 其目标是在完成多楼层环境中所有取送货任务并最终返回仓库的前提下, 最小化总行驶距离. 事实上, 所有具有多个子区域且区域间连接相对稀疏的 PDP 变体均可被视为 MF-PDP. 除传统 PDP 的优先级约束外, MF-PDP 还涉及子区域划分导致的特殊拓扑结构以及子区域间稀疏连接导致的特殊访问限制.

MF-PDP 的完整节点集定义为: $X=\{x_0\}\cup P\cup D\cup E$. 其中, x_0是仓库节点, 对应车辆的起始与终止位置; $P=\{x_1,\ldots,x_n\}$和 $D=\{x_{n+1},\ldots,x_{2n}\}$ 分别表示取货节点集和送货节点集, 且对任意 $x_i\in P$, 其对应的送货节点为 $x_{i+n}\in D$; $E=\{x_{2n+1},\ldots,x_{2n+m}\}$表示电梯节点集, 其中$m$为电梯节点数量. 对于任意节点$x_i\in\{x_0\}\cup P\cup D$, 均关联一个楼层属性$f_i$, 用于指示该节点所属楼层. 电梯节点$x_i\in E$不定义楼层属性, 其作为跨楼层的连接节点, 假定与所有非电梯节点连通.

为简化问题建模, 假设车辆具有无限容量. 实际场景中, 一次完整的运输任务通常默认满足车辆的载重限制, 因此对于车辆的无限容量假设是合理的. 此外, 该假设便于研究聚焦于稀疏跨层拓扑对求解性能的影响, 排除了容量约束与电梯动态调度等因素引入的额外复杂性, 以便更清晰地评估所提编码与解码机制在应对拓扑挑战方面的有效性. 设 $y_{ij}\in\{0,1\}$为二元决策变量, 表示车辆是否从节点x_i直接行驶至节点x_j, 并令B_i表示车辆到达x_i时的累计行驶距离. 定义节点x_i与x_j之间的行驶距离为$d(i,j)$, 其计算方式如下:

$$d(i,j)=\begin{cases}||x_i-x_j||, f_i=f_j\\ \min_{e\in[2n+1,2n+m]}(||x_i-x_e||+\\ \quad ||x_j-x_e||),\ f_i\neq f_j\end{cases}. \tag{1}$$

其中$\|\cdot\|$表示欧式距离.

基于上述定义, MF-PDP 的目标函数可表示为:

$$\min \sum_{i\in X}\sum_{j\in X} d(i,j)\cdot y_{ij}. \tag{2}$$

并满足以下约束:

$$\sum_{j=0}^{2n+m} y_{ij}=1,\ i\in[0,2n], \tag{3}$$

$$\sum_{i=0}^{2n+m} y_{ij}=1,\ j\in[0,2n], \tag{4}$$

$$y_{ij}\in\{0,1\},\ i,j\in[0,2n+m], \tag{5}$$

$$B_j\geqslant B_i+d(i,j)-\phi(1-y_{ij}),\ i,j\in[0,2n+m], \tag{6}$$

$$B_{i+n}\geqslant B_i+d(i,i+n),\ i\in[1,n], \tag{7}$$

$$|f_i-f_j|\leqslant\phi(1-y_{ij}),\ i,j\in[0,2n], \tag{8}$$

$$y_{ij}=0, i,j\in[2n+1,2n+m],\ i\neq j. \tag{9}$$

其中, ϕ为一个足够大的常数. 约束 (3) 和 (4) 用于保证每个取货节点、送货节点及仓库节点均被恰好访问一次 (电梯节点$x_i\in E$允许被多次访问, 故不受该约束限制). 约束 (5) 定义了二元决策变量的取值范围. 约束 (6) 通过对累计行驶距离变量B_i施加单调递增条件以消除子回路. 当$y_{ij}=1$时, $B_j\geqslant B_i+d(i,j)$保证路径上距离严格递增, 使回路无法满足该单调性要求. ϕ取所有节点对间最大可能距离的上界, 以保证$y_{ij}=0$时约束自动松弛. 约束 (7) 用于保证每个送货节点在其对应的取货节点访问完成之后才能被访问. 约束 (8) 用于限制车辆仅能在同一楼层的节点之间进行直接行驶, 从而保证任意跨楼层移动必须至少经过一个电梯节点. 由于电梯节点不定义楼层属性, 不受该约束限制, 因此车辆可经由电梯节点实现跨楼层转移. 约束 (9) 用于禁止在不访问其他节点的情况下在电梯节点之间进行连续切换, 以避免产生不必要的无效移动.

MF-PDP 与经典 PDP 的根本差异并非简单增加楼层属性与电梯节点, 而是体现在以下三个层面. 在距离度量上, 经典 PDP 中任意两节点间的行驶代价由欧氏距离直接确定, 而 MF-PDP 中不同楼层节点之间无法直接到达, 必须经由电梯节点中转, 行驶距离为到达电梯与离开电梯两段路径代价之和, 且实际经由哪个电梯受路径整体规划的影响, 使得跨楼层行驶代价依赖具体的路径选择, 无法仅由节点坐标唯一确定.

在约束结构上, 经典 PDP 中任意两节点均可直接转移, 求解只需确定满足取送先后的访问顺序; 而约束 (8) 要求跨楼层移动必须经电梯中转, 访问顺序与跨层可达性相互耦合, 取送对分布于不同楼层时, 访问顺序、跨层切换时机与中转电梯选择三者相互制约. 在解空间上, 电梯节点允许重复访问, 解路径长度不再固定, 每次跨楼层移动均引入中转电梯选择这一额外决策维度, 使解空间呈组合级增长. 经典 PDP 是 NP-hard 问题, 且是 MF-PDP 退化为单楼层且无电梯时的特例. 结合上述分析, MF-PDP 是经典 PDP 在分层稀疏拓扑下的结构化推广, 求解复杂性进一步增加.

3 DRL-SCFO 方法

针对 MF-PDP 中复杂拓扑结构所带来的挑战, 本文提出了一种基于深度强化学习的稀疏跨层路径优化 (DRL-SCFO) 方法. 该方法采用编码器-解码器架构, 其中编码器为双视域空间注意力编码器 (DSAenc), 解码器为潜在规划-实际执行双阶段解码器 (PDSdec). DRL-SCFO 的整体架构如图 2 所示. 下文将依次介绍 DSAenc 与 PDSdec 的设计思想与实现细节, 并在本节最后给出相应的训练算法.

3.1 双视域空间注意力编码器

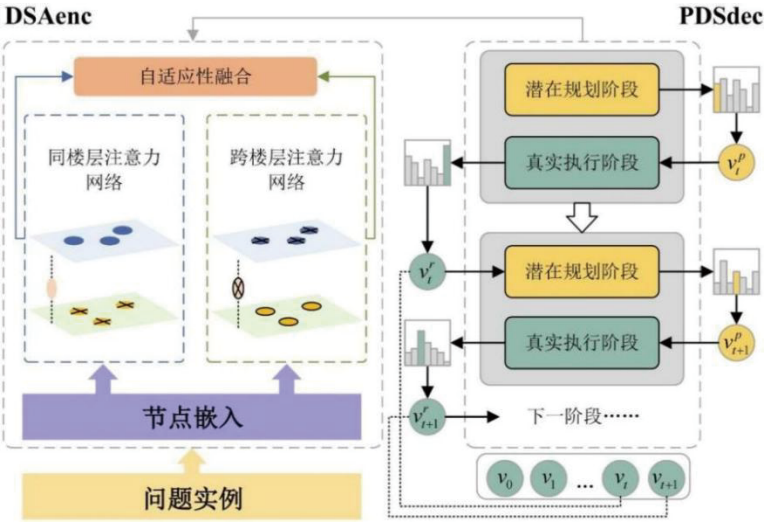

图 2 DRL-SCFO 方法整体框架

在节点特征初始化阶段, 为增强取货节点的表征能力, 将每个取货节点x_i的特征向量与其对应送货节点x_{i+n}的特征进行拼接, 得到$x_i = [x_i; x_{i+n}]$. 对于其它节点, 则保留其原始特征向量. 随后, 将所有节点特征统一投影至d_h维的潜在空间中, 以获得初始节点嵌入$\boldsymbol{h}_i^0$, 并将其作为编码器的输入. DSAenc 旨在同时建模同一个楼层内与跨楼层间的空间结构关系. 为此, DSAenc 采用双分支结构, 分别由一个楼层内视域注意力网络和一个跨楼层视域注意力网络构成. 两个分支均包含L层具有独立参数的堆叠子网络. 每个视域网络均基于 Transformer 编码器结构构建[22], 并通过多头注意力 (Multi-Head Attention, MHA) 机制与前馈网络对节点表示进行逐层更新. 两个视域网络不同之处在于, 它们的注意力子层中采用了不同的掩码策略, 分别记为$\boldsymbol{M}^{\text{intra}}$和$\boldsymbol{M}^{\text{inter}}$, 以约束各自的注意力作用范围. 这些掩码分别在公式 (10) 和 (11) 中逐一给出定义:

$$\boldsymbol{M}_{ij}^{\text{intra}} = \begin{cases} 0, \text{ if } f_i = f_j \text{or} x_j \in E \\ -\infty, \text{ otherwise} \end{cases} \tag{10}$$

$$\boldsymbol{M}_{ij}^{\text{inter}} = \begin{cases} 0, \text{ if } f_i \neq f_j \\ -\infty, \text{ otherwise} \end{cases} \tag{11}$$

上述两种掩码将节点间的交互关系按楼层归属划分为两个互斥的子集, 分别约束对应分支的注意力作用范围. MF-PDP 中同楼层节点之间直接可达, 其行驶代价由平面欧氏距离唯一确定; 跨楼层节点必须经由电梯中转, 其实际代价取决于中转电梯的选择且具有路径依赖性. 若采用单一全连接注意力对所有节点统一建模, 这两类性质迥异的关系在同一个 softmax 归一化空间中混合计算, 模型被迫用同一组投影参数同时拟合直接可达的局部交互与需要中转的跨层依赖, 增加了特征学习的难度.

因此, 本文构造了如图 3 所示的网络结构. 其中, $\boldsymbol{M}^{\text{intra}}$将楼层内分支中每个节点的信息传递限制在同楼层节点与电梯节点之间 (图中同色节点与电梯节点间的连线), softmax 归一化的作用范围限定在局部可达节点集合上, 使注意力权重专注于刻画局部可达节点间的相对重要性; $\boldsymbol{M}^{\text{inter}}$将跨楼层分支中每个节点的信息传递限制在不同楼层节点之间 (图中不同色节点间的连线), 使跨层依赖关系不受楼层内交互的干扰而获得充分表征. 两个分支各自拥有独立的投影参数, 可分别学习适合各自关系类型的特征变换, 降低了单一参数空间同时拟合两类异质关系的优化难度. 经过L层编码后, 为后续的特征融合提供具有区分性的空间表征. 下面以单头注意力为例, 说明上述掩码在注意力计算中的具体使用方式.

在单头注意力中, 第l层的注意力输出是基于前一层隐藏表示$\boldsymbol{H}^{l-1}$计算得出的:

$$\text{Attention}(\boldsymbol{Q}_i, \boldsymbol{K}, \boldsymbol{V}; \boldsymbol{M}_i) = \text{soft}\max\left(\frac{\boldsymbol{Q}_i^T \boldsymbol{K}}{\sqrt{d_h}} + \boldsymbol{M}_i\right)\boldsymbol{V}^T, \tag{12}$$

$$\boldsymbol{Q}_i = \boldsymbol{W}_{Q,e}^l h_i^{l-1},\ \boldsymbol{K} = \boldsymbol{W}_{K,e}^l \boldsymbol{H}^{l-1},\ \boldsymbol{V} = \boldsymbol{W}_{V,e}^l \boldsymbol{H}^{l-1}. \tag{13}$$

其中, $\boldsymbol{W}_{Q,e}^l$, $\boldsymbol{W}_{K,e}^l$和$\boldsymbol{W}_{V,e}^l$是第l层的可学习投影矩阵; 向量$\boldsymbol{h}_i^{l-1}$表示来自第$l-1$层的节点x_i的嵌入; 向量$\boldsymbol{Q}_i$以及矩阵$\boldsymbol{K}$和$\boldsymbol{V}$分别代表查询、键和值.

图 3 描述了 DSAenc 双分支架构中的信息传递过程. 其中, 每个分支网络均由L层堆叠子网络构成, 每一个堆叠子网络均包含一个应用了$\boldsymbol{M}^{\text{intra}}$或$\boldsymbol{M}^{\text{inter}}$的 MHA 子层和一个紧随其后的逐节点 FF 子层. $\boldsymbol{M}^{\text{intra}}$使得楼层内视域注意力网络中的 MHA 子层只关注那些与节点x_i位于同一楼层的节点或电梯节点, 从而使节点x_i能够聚合那些由节点x_i出发直接可达的节点的信息. 相对地, $\boldsymbol{M}^{\text{inter}}$使得跨楼层视域注意力网络的 MHA 子层只关注和节点x_i不同楼

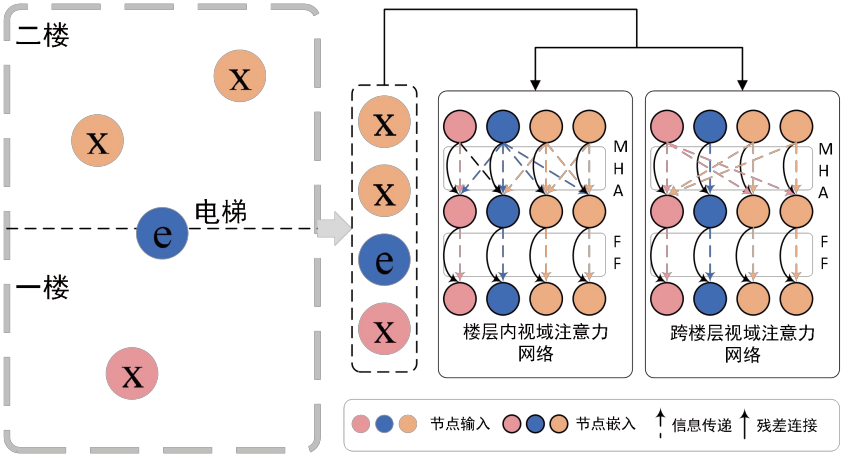

图 3　DSAenc 的双分支架构

层的节点, 从而有效聚合节点x_i无法直接访问的区域的信息. 通过上述两种掩码机制, 双分支注意力网络在编码阶段分别建模了细粒度的楼层内交互关系与长距离的跨楼层空间依赖关系. 随后, MHA 子层的输出进一步经由前馈网络 (Feed Forward, FF) 处理. 如图 3 所示, MHA 和 FF 所构成的每个子层均引入了残差连接[23] 和批量归一化 (Batch Normalization, BN)[24] 以稳定训练过程并增强特征表达能力.

然而, 在实际问题中, 当楼层数量较大时, 两个视域注意力网络所提取的信息可能存在不平衡现象. 为缓解该问题, 本文在编码模块中引入了一种逐维自适应融合机制, 用以动态整合来自不同空间视域的注意力输出. 具体而言, 定义两个可学习参数向量$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{d_h})$, $\boldsymbol{\gamma} = (\gamma_1, \gamma_1, \ldots, \gamma_{d_h})$, 并通过 softmax 函数对每个嵌入维度$k \in \{1, \ldots, d_h\}$进行归一化, 如公式 (14) 和 (15) 所示:

$$\beta_k = \mathrm{softmax}(\frac{e^{\beta_k}}{e^{\beta_k+\gamma_k}}), \tag{14}$$

$$\gamma_k = \mathrm{softmax}(\frac{e^{\gamma_k}}{e^{\beta_k+\gamma_k}}). \tag{15}$$

随后, 通过对经过L个编码层后的两个注意力网络的输出$\boldsymbol{h}_i^{L,\mathrm{intra}}$和$\boldsymbol{h}_i^{L,\mathrm{inter}}$进行逐维加权求和, 可以得到节点$x_i$的最终嵌入$\boldsymbol{h}_i^L$:

$$\boldsymbol{h}_i^L = \boldsymbol{\beta} \odot \boldsymbol{h}_i^{L,\mathrm{intra}} + \boldsymbol{\gamma} \odot \boldsymbol{h}_i^{L,\mathrm{inter}}. \tag{16}$$

其中$\odot$表示逐元素乘法. 可学习参数向量$\boldsymbol{\beta}$和$\boldsymbol{\gamma}$初始化为零向量, 经 softmax 归一化后各维度初始权重为$\boldsymbol{\beta}_k = \boldsymbol{\gamma}_k = 0.5$, 即等价于对两分支输出取均值, 避免在训练初期引入人为偏置. 在训练过程中, $\boldsymbol{\beta}$和$\boldsymbol{\gamma}$随网络参数共同更新, 逐步学习各嵌入维度对楼层内与跨楼层信息的融合比例.

3.2 潜在规划-实际执行双阶段解码器

PDSdec 将每一步决策分解为两个阶段: 全局潜在规划阶段 (简称“规划阶段”) 与局部实际执行阶段 (简称“执行阶段”). 在规划阶段, 解码器在时间步t从全局候选集$\mathcal{V}_t$中选择一个规划节点v_t^p. 其中, $\mathcal{V}_t$为那些与当前节点之间满足取送优先级等 PDP 基本约束的节点构成的集合. 所选规划节点v_t^p的嵌入表征$\boldsymbol{h}_t^p$被传递至执行阶段以提供跨楼层的全局引导. 随后, 在该引导信息的作用下, 执行阶段在时间步t从局部可行集$\mathcal{F}_t$中选择实际执行节点v_t^r. 其中, $\mathcal{F}_t$由当前节点直接可达的节点构成, 并需满足 MF-PDP 的全部可行性约束 (即公式 (3)-(9)). 在下一时间步$t+1$, 规划阶段节点的选择将结合上一时间步执行阶段所选节点v_t^r的嵌入表征$\boldsymbol{h}_t^r$进行调整, 从而使两阶段在“规划—执行—再规划”的循环中形成相互引导、逐步修正的协同关系, 如图 4 所示. 此外, 上一时间步中所选规划节点和执行节点的嵌入表征分别记为$\boldsymbol{h}_{t-1}^p$和$\boldsymbol{h}_{t-1}^r$. 为使当前决策显式利用上一时间步两阶段的选择结果, 本文将$\boldsymbol{h}_{t-1}^p$与$\boldsymbol{h}_{t-1}^r$分别引入时间步t的规划阶段与执行阶段的上下文中, 从而为后续解码提供更充分的历史信息. 因此, 两个阶段对应的上下文嵌入$\boldsymbol{h}_t^{p,c}$和$\boldsymbol{h}_t^{r,c}$分别由式 (17) 和 (18) 定义, 并且它们被作为规划阶段与执行阶段在时间步t的上下文输入:

$$\boldsymbol{h}_t^{p,c} = \mathrm{Concat}(\overline{\boldsymbol{h}}_G, \boldsymbol{h}_{t-1}^r, \boldsymbol{h}_{t-1}^p), \tag{17}$$

$$\boldsymbol{h}_t^{r,c} = \mathrm{Concat}(\overline{\boldsymbol{h}}_F, \boldsymbol{h}_t^p, \boldsymbol{h}_{t-1}^r). \tag{18}$$

其中$\overline{\boldsymbol{h}}_G$表示图嵌入, 由所有节点嵌入取平均得到; $\overline{\boldsymbol{h}}_F$表示楼层嵌入, 由与当前节点处于同一楼层的节点嵌入平均得到. 规划阶段的上下文中引入了$\boldsymbol{h}_{t-1}^r$,

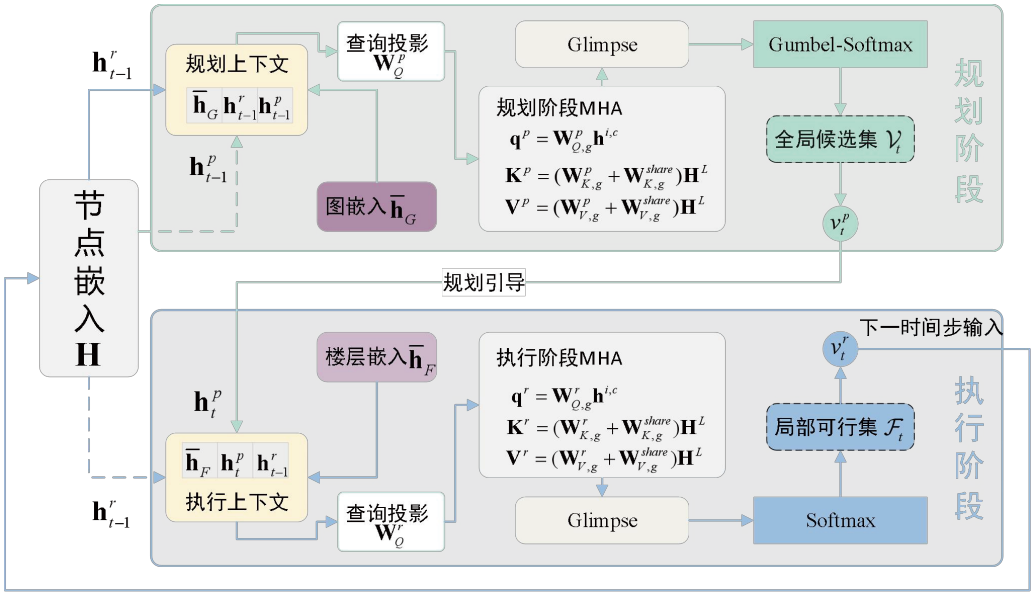

图 4 PDSdec 的解码流程

从而使全局规划能够依据上一时间步的实际执行结果及时修正; 相应地, 执行阶段的上下文中引入了$\boldsymbol{h}_t^p$, 以使得局部选择在满足可行性约束的同时, 能够响应跨楼层的全局引导, 从而做出更合理的节点选择. 通过规划与执行两阶段的逐步交互, PDSdec 能够更好地适应 MF-PDP 中楼层间稀疏连接这一特殊拓扑结构所带来的挑战.

接着, 将公式 (17) 或 (18) 得到的上下文嵌入作为线性投影层的输入, 以生成对应阶段的查询向量. 该查询向量进一步送入 MHA 模块, 得到相应的 glimpse 向量, 并用于指导当前阶段的节点选择. 如图 4 所示, 执行阶段和规划阶段在解码结构上保持一致, 其 MHA 模块在参数层面部分共享. 设$s \in \{p, r\}$表示当前阶段 (其中p表示规划阶段, r表示执行阶段), 则阶段s的 glimpse 向量$\boldsymbol{h}_g^s$可统一表示为:

$$\boldsymbol{h}_g^s = \mathrm{MHA}\begin{pmatrix} \boldsymbol{W}_{Q,g}^s \boldsymbol{h}^{s,c} \\ (\boldsymbol{W}_{K,g}^s + \boldsymbol{W}_{K,g}^{share})\boldsymbol{H}^L \\ (\boldsymbol{W}_{V,g}^s + \boldsymbol{W}_{V,g}^{share})\boldsymbol{H}^L \end{pmatrix}. \tag{19}$$

其中, $\boldsymbol{W}_{Q,g}^s \in \mathbb{R}^{d_h \times d_k}$和$\boldsymbol{W}_{V,g}^s \in \mathbb{R}^{d_h \times d_k}$分别为阶段$s$的查询、键和值投影矩阵, $\boldsymbol{W}_{K,g}^{share}$和$\boldsymbol{W}_{V,g}^{share}$为两个阶段共享的键和值投影参数. 阶段特定参数用于保留规划与执行在决策目标上的差异, 共享投影用于约束两阶段学习一致的底层语义特征, 从而促进规划阶段与执行阶段之间的协同. 每个阶段中最终使用的键、值投影矩阵都是由阶段特定投影与共享投影之和构成的. 随后, 上述查询、键和值的投影共同作为 MHA 模块的输入, 为每个阶段生成对应的 glimpse 向量$\boldsymbol{h}_g^s$. 利用该 glimpse 向量, 并参照 AM[5] 和 Heter-AM[6] 的解码方式, 分别计算规划阶段和执行阶段的节点选择概率分布, 并据此从候选集$\mathcal{V}_t$和可行集$\mathcal{F}_t$中采样得到规划节点v_t^p与执行节点v_t^r. 在采样策略上, 执行阶段采用标准 softmax, 其动作概率直接参与策略梯度计算, 可获得有效训练信号. 规划阶段的输出为不进入最终路径序列的离散中间变量, 使用标准采样会导致梯度在此处断裂, 故采用 Gumbel-Softmax[25] 进行可微近似, 使路径成本的训练信号能够回传至规划阶段参数, 实现两阶段的端到端联合优化.

上述两阶段决策结构使得 PDSdec 的策略空间相比单阶段解码器得到了拓展. 设时间步t的解码状态为s_t, 单阶段解码器将s_t直接映射为局部可行集$\mathcal{F}_t$上的概率分布$P(a_t|s_t)$, $a_t \in \mathcal{F}_t$. PDSdec 中, 规划节点v_t^p不进入最终路径序列, 其在策略中的角色等价于潜变量, 记为$z_t = v_t^p, v_t^p \in \mathcal{V}_t$. 据此, PDSdec 的策略可形式化为潜变量增强的分层结构:

$$P_{\mathrm{dual}}(a_t|s_t) = \sum\nolimits_{z_t \in V_t} P_p(z_t|s_t) \cdot P_r(a_t|s_t, z_t), a_t \in \mathcal{F}_t. \tag{20}$$

其中$P_p(z_t|s_t)$为规划策略在$\mathcal{V}_t$上的概率分布, $P_r(a_t|s_t, z_t)$为以z_t为条件的执行策略在$\mathcal{F}_t$上的概率分布. 当$P_r(a_t|s_t, z_t)$对所有$z_t \in \mathcal{V}_t$相同时, 上式退化为与z_t无关的单一分布, 等价于单阶段策略; 而当不同的z_t通过公式 (18) 改变执行上下文从而导致不同的条件分布时, P_{dual}成为多个不同分布的凸组合, 能够表达单一分布无法覆盖的多模态动作偏好. 因此, PDSdec 的策略空间严格包含单阶段解码器的策略空间.

在 MF-PDP 中, $\mathcal{F}_t$受约束 (8) 限制仅包含当前楼层节点与电梯节点, 而$\mathcal{V}_t$还包含其他楼层的取送货节点, 满足$\mathcal{F}_t \subset \mathcal{V}_t$. 上述策略空间的扩展在此具有明确的结构意义. 当$z_t \in \mathcal{V}_t \backslash \mathcal{F}_t$为其他楼层节点时, 其嵌入$\boldsymbol{h}_t^p$进入执行上下文, 使$P_r(\cdot|s_t, z_t)$倾向于为电梯节点分配较高概率以引导车辆向该跨楼层目标移动. 当$z_t \in \mathcal{F}_t$为当前楼层节点时, $P_r(\cdot|s_t, z_t)$倾向于直接选择同楼层取送货节点. 这种依据不同全局目标对局部动作产生差异化选择倾向的能力, 是单阶段解码器对同一状态只能输出固定分布所无法实现的. PDSdec 将全局规划与局部执行显式解耦, 使每一步局部动作都与明确的全局目标相关联, 从而有效缓解了上述问题.

令序列$\pi = (v_0^r, v_1^r, \ldots, v_t^r)$表示由 PDSdec 生成的执行节点访问序列, 则对于给定的问题实例X, 其解码序列的生成概率可分解为:

$$\begin{aligned} P(\pi|X) = {} & P_\theta(v_0^p|X) P_\theta(v_0^r|v_0^p, X) \cdot \\ & \prod_{t=1}^{T} P_\theta(v_t^p|v_{1:\,t-1}^r, v_{1:\,t-1}^p, X) \\ & P_\theta(v_t^r|v_{1:\,t-1}^r, v_t, X). \end{aligned} \tag{21}$$

其中, θ表示模型中所有的可学习参数.

3.3　算法模型训练

DRL-SCFO 的训练过程采用基于 REINFORCE[26] 的策略梯度方法. 在每个训练实例上, 模型在当前策略下逐步构造一条完整路径, 并以其总成本作为奖励, 用于指导策略参数更新. 而前述的双阶段解码机制仍属于参数化随机策略, 可在 REINFORCE 策略梯度框架下进行优化. 其策略梯度仍为无偏估计量, 在适当的学习率调度下可保证策略参数收敛至局部最优[26]. 整体训练流程如算法 1 所示, 主要包括规划阶段与执行阶段的逐步采样、greedy rollout 基线的

计算以及网络参数的更新. 对于给定的实例X, 策略梯度的估计形式如下:

$$\nabla\mathcal{L}(\theta|X)=E_{P_\theta(\pi|X)}[(c(\pi|X))-b(X)\nabla\log P_\theta(\pi|X)] \tag{22}$$

其中, $c(\pi|X)$表示路径π在实例X上的总成本 (即路径长度), $b(X)$表示 greedy rollout 基线策略在实例X上得到的路径成本, 用于降低梯度估计方差. 对于每个训练批次, 本文采用 greedy rollout 作为基线策略. 在 greedy rollout 基线机制下, 当当前策略在实例X上得到的解优于基线解时, 基线策略将更新为当前策略. 与传统 Actor-Critic 方法中引入的 Critic 基线[5] 相比, 该基线无需额外训练价值网络, 从而在实际训练过程中具有更好的稳定性. 模型参数θ通过基于该基线的 REINFORCE 梯度估计进行更新. 在每个训练 epoch 结束后, 仅当当前策略在性能上显著优于基线策略时, 才对基线进行更新. 这一显著性统计检验是通过对当前策略与 greedy rollout 基线在路径总成本上的性能差异进行配对 t 检验来完成的, 验证对象为一个包含 10 000 个实例的独立验证集, 显著性水平设为$\alpha=0.05$.

算法 1 基于REINFORCE的DRL-SCFO训练算法

输入: 训练集X, 训练轮数E, 每轮批次数S, 批量大小B, 策略参数θ, 基线策略参数θ^{BL}, 显著性水平α

初始化当前策略网络参数θ, θ^{BL}
for $epoch=1$ to E
  for $step=1$ to S
    for $k=1$ to B
      $X_k\leftarrow\mathrm{RandomInstance}(X)$ //随机采样
      初始化空路径π_k
      $t\leftarrow 0$
      while 解 π_k未完成 do
        $v_t^p\sim P_{\theta,t}^p(v_{1:t-1}^r, v_{1:t-1}^p, X_i)$ //Gumbel-Softmax
        $v_t^r\sim P_{\theta,t}^r(v_{1:t-1}^r, v_{1:t}^p, X_i)$ //Softmax
        $\pi_t\leftarrow[\pi_{t-1}, v_t^r]$
        $t\leftarrow t+1$
      end while
      $\pi_k^{BL}\leftarrow P_{\theta^{BL}}(X_k)$ // 基线策略采样
    end for
    使用公式(22)计算得到$\nabla\mathcal{L}$
    $\theta\leftarrow\mathrm{Adam}(\theta,\nabla\mathcal{L})$
  end for
  If $\mathrm{OneSidedPairedTTest}(P_\theta, P_{\theta^{BL}})<\alpha$ then
    $\theta^{BL}\leftarrow\theta$
  end if
end for

输出: 可行解 π

4 实验结果与分析

4.1 实验设置

在实验中, 所有节点的二维空间坐标均在单位正方形内随机生成, 即服从二维均匀分布$\mathcal{U}[0,1]^2$, 这与相关研究工作[6,14] 保持一致. 节点的楼层属性在整数区间$[1, f]$上独立采样, 服从离散均匀分布. 为准确刻画 MF-PDP 的问题拓扑, 本文采用三元组(n, m, f)描述问题规模, 其中n表示取货与送货节点的总数 (即对应$n/2$个取送货任务), m表示电梯节点数量, f表示楼层数量. 后续实验分别在三种不同规模的 MF-PDP 实例上进行, 即 (20, 3, 3)、(50, 4, 4) 和 (100, 5, 5), 分别对应小规模、中规模和大规模的 MF-PDP 问题. 对于不同数据分布与问题规模的组合, 我们分别生成 10 000 个验证实例和 10 000 个测试实例用于性能评估.

在评估阶段, 本文分别报告 greedy 与 sampling-1280 两类推理设置下的结果. 其中, greedy 表示在每一步解码时选择概率最大的动作, sampling-1280 表示对每个测试实例采样 1280 条完整路径, 并选取其中代价最小的路径作为最终解. 为保证采样预算的一致性, 所有基于 DRL 的方法在 sampling-1280 设置下均采用相同采样次数. 后续实验表格通过 Setting 列列出各方法对应的求解或推理设置.

4.2 与基线方法的比较

本节将 DRL-SCFO 与多种基线方法进行对比分析, 包括两种传统优化方法以及四种基于 DRL 的方法. 其中, 两种传统方法分别为 Gurobi[27](一种基于分支定界 (branch-and-cut) 的商业优化求解器) 和 OR-Tools[28] (一种开源的启发式求解框架). 四种基于 DRL 的基线方法包括 AM[5]、Heter-AM[6]、MVMoE[7] 和 RRNCO[11], 相关方法已在研究现状分析章节中进行介绍. 对于 Gurobi[27] 和 OR-Tools[28], 本文对单个实例设置了 1 000 秒的求解时间限制, 并仅在小规模 (20, 3, 3) 实例上将两者纳入对比, 其中 Gurobi 在时限内求得的解作为解质量的参考基准. 在 (50, 4, 4) 和 (100, 5, 5) 规模实例上, 由于 MF-PDP 中电梯节点允许重复访问且约束 (8)(9) 引入了跨层可达性限制, Gurobi 的分支搜索空间急剧膨胀, OR-Tools 则难以在其启发式框架下构造满足全部跨层约束的可行解, 两者均未能在时限内返回结果. 这一现象并非 MF-PDP 所特有, 在经典 PDP 中精确求解器同样仅能处理小规模实例[6]. MF-PDP 更为复杂的

约束结构进一步加剧了这一局限性. 基于上述原因, 本文仅对它们在小规模实例上的实验结果进行了统计. 表 1 汇总了各方法在不同实例规模下的平均目标函数值 (Obj.), 与所有对比方法能找到的最优解之间的性能差距 (Gap), 以及对应的所有实例总的推理时间.

表 1　DRL-SCFO 与基线模型的性能比较 (DRL 方法含 greedy 与 sampling-1280 两种设置)

Method	Setting	(20, 3, 3)			(50, 4, 4)			(100, 5, 5)		
		Obj.	Gap	Time	Obj.	Gap	Time	Obj.	Gap	Time
Gurobi	MIP solver	7.74	1.70%	413.62s	—	—	—	—	—	—
OR-Tools	heuristic solver	7.85	3.15%	784.25s	—	—	—	—	—	—
AM	greedy	8.44	10.91%	**0.81s**	15.39	11.68%	**1.72s**	24.15	9.92%	**3.89s**
Heter-AM	greedy	8.44	10.91%	1.04s	15.38	11.61%	1.93s	24.38	10.97%	3.94s
MVMoE	greedy	8.43	10.78%	1.11s	15.37	11.54%	2.25s	24.07	9.56%	2.65s
RRNCO	greedy	8.08	6.17%	1.12s	14.83	7.54%	2.40s	23.79	8.28%	8.84s
DRL-SCFO	greedy	7.90	3.81%	1.04s	14.62	6.04%	2.70s	23.36	6.33%	7.24s
AM	sampling-1280	7.93	4.20%	126.36s	14.22	3.19%	303.60s	22.33	1.64%	527.21s
Heter-AM	sampling-1280	7.92	4.07%	404.57s	14.20	3.05%	318.42s	22.22	1.14%	568.43s
MVMoE	sampling-1280	7.92	3.94%	463.55s	14.21	3.05%	352.04s	22.28	1.41%	711.17s
RRNCO	sampling-1280	7.73	1.58%	478.64s	13.81	0.15%	471.69s	22.05	0.36%	846.51s
DRL-SCFO	sampling-1280	**7.61**	**0.00%**	483.80s	13.79	**0.00%**	426.21s	**21.97**	**0.00%**	756.27s

如表 1 所示, DRL-SCFO 在两种评估设置下的解质量均优于各基线方法. 在 (20,3,3) 规模实例上, DRL-SCFO 的目标函数值较次优方法 RRNCO 降低了 2.23%, 较 AM、Heter-AM 和 MVMoE 降低了约 6.3%~6.8%. 附录中还对一个 (20, 3, 3) 实例上的注意力权重分布和路径决策结果进行了可视化分析. 在 (50,4,4) 和 (100,5,5) 规模实例上, DRL-SCFO 同样取得了最低的目标函数值, 均优于 RRNCO 及其他对比方法. 上述结果表明, DRL-SCFO 的解质量优势源于模型结构本身, 而非依赖高次采样. 在小规模实例上, 与两种传统方法相比, DRL-SCFO(g) 的目标函数值与 Gurobi 较为接近, 且优于 OR-Tools.

在 sampling-1280 设置下, 所有 DRL 方法均采用相同的采样次数, DRL-SCFO 在各规模上仍取得了最优的目标函数值, 优于次优方法 RRNCO. 在小规模实例上, DRL-SCFO(s) 的目标函数值优于 Gurobi 和 OR-Tools. 随着问题规模增大, 各方法之间的 Gap 呈现逐步缩小趋势, 这可能与大规模实例中单个楼层内节点数量增加有关, 使得其他 DRL 方法受跨楼层稀疏拓扑的影响相对减弱.

在推理时间方面, 由于 PDSdec 引入了两阶段决策过程, DRL-SCFO 的推理时间略高于 AM 等单阶段解码方法, 但与 RRNCO 处于同一量级. 在 sampling-1280 设置下, DRL-SCFO 的推理时间在 (50,4,4) 和 (100,5,5) 规模上甚至低于 RRNCO. 综合来看, DRL-SCFO 在解质量显著提升的前提下, 所引入的额外开销并不会成为主要瓶颈.

4.3　消融实验

为了进一步分析 DRL-SCFO 中编码器和解码器的有效性, 本小节将 DSAenc 和 PDSdec 分别替换为 AM 的编码器和解码器 (即 AMenc 和 AMdec), 来进行对应的消融实验. 所有消融模型均在相同的超参数配置下从头训练, 并采用与表 1 一致的 greedy 和 sampling-1280 推理设置进行评估, 以确保实验结果的可比性. 各模型在测试集上的性能结果汇总于表 2 中.

表 2　DSAenc 与 PDSdec 的消融实验

Method	Setting	(20, 3, 3)		(50, 4, 4)		(100, 5, 5)	
		Obj.	Gap	Obj.	Gap	Obj.	Gap
AMenc+PDSdec	greedy	8.16	7.24%	15.16	9.94%	23.84	8.51%
DSAenc+AMdec	greedy	8.00	5.12%	14.77	7.08%	23.72	7.94%
Full model	greedy	7.90	3.81%	14.62	6.04%	23.36	6.33%
AMenc+PDSdec	sampling-1280	7.86	3.22%	14.12	2.41%	22.20	1.01%
DSAenc+AMdec	sampling-1280	7.64	0.41%	13.84	0.38%	22.15	0.82%
Full model	sampling-1280	**7.61**	**0.00%**	**13.79**	**0.00%**	**21.97**	**0.00%**

如表 2 所示, DRL-SCFO 在两种解码策略下的目标函数值均低于各消融变体, 表明所提出的 DSAenc 和 PDSdec 在整体模型中发挥了积极作用. 具体而言, 在编码器消融实验中, 采用 AMenc 替换 DSAenc 后, 目标函数值相较完整模型增加约 1.0% 到 3.5%, 表明 DSAenc 在刻画 MF-PDP 所固有的复杂空间关系方面更具优势. 对于采用 AMdec 替换 PDSdec 的模型, 其目标函数值相较完整模型至少增加约 0.38%, 表明在解码阶段, 仅依赖局部单跳邻域

信息的 AMdec 难以充分应对 MF-PDP 的全局规划挑战, 而 PDSdec 所引入的两阶段决策机制有助于增强模型的全局路径规划能力. 此外, 对比结果可以发现, 其差异显著大于表 2 中任一单一组件消融实验所带来的性能变化. 这说明, 所提出的编码器和解码器的协同作用能够进一步激发彼此的潜力, 从而促成了 DRL-SCFO 优异的整体性能表现. 这一现象源于两个模块在信息链路上的功能互补. DSAenc 的 inter 分支经过L层编码后使节点嵌入显式携带跨楼层依赖信息, PDSdec 的规划阶段在全局候选集$\mathcal{V}_t$上选择规划节点后, 通过公式 (18) 将其嵌入注入执行上下文, 从而将全局规划意图转化为局部决策的引导信号. 当编码器退化为 AMenc 时, 节点嵌入不具备楼层区分性, 规划阶段无法从中提取有效的全局视野. 当解码器退化为 AMdec 时, DSAenc 编码的双视域信息缺乏传递至局部决策的通道. 因此, 两个模块缺失任一均会削弱另一模块的效能, 完整模型的性能提升显著大于各单一组件消融所带来的变化.

4.4 泛化能力分析

为系统评估 DRL-SCFO 的泛化能力, 本节从多个维度进行实验验证, 包括跨数据分布与问题规模的泛化、向传统 PDP 退化场景的泛化, 以及面向真实多楼层场景的分布外泛化. 上述实验中, 各方法均使用在均匀随机分布上训练完成的模型权重, 不进行重新训练或微调. 具体而言, 所采用的数据分布为截断二维正态分布, 该分布的均值为$\mu = (0.5, 0.5)$, 标准差为$\sigma = 0.15$, 并截断于正方形区域$[0, 1]^2$内; 所采用的问题规模为 (50, 4, 4) 和 (100, 5, 5) 这两种更大的问题规模, 以及一种 (20, 0, 1) 的退化情形. 相应的实验结果汇总于表 3 中. 需要指出的是, (20, 0, 1) 的设置对应了一种不存在电梯且所有节点均位于同一楼层的特殊 MF-PDP, 实际上就等价于传统的 PDP. 因此, 这一实验可进一步评估 DRL-SCFO 在传统 PDP 场景下的泛化能力. 如表 3 所示, DRL-SCFO 在 greedy 和 sampling-1280 两种解码策略下均取得了较低的目标函数值. 在 (50, 4, 4) 和 (100, 5, 5) 两种更大规模实例上, 即使在数据分布发生变化且问题规模增大的情况下, DRL-SCFO 的目标函数值较 RRNCO 在 greedy 设置下仍分别降低了 1.3% 和 7.8%.

此外, 在 (20, 0, 1) 规模问题上的实验结果表明, DRL-SCFO 在向传统 PDP 场景迁移时仍能够取得较好的求解效果. 值得注意的是, RRNCO 在该退化场景下性能显著下降, 其目标函数值远高于其他方法, 这表明 RRNCO 的边特征编码机制对多楼层拓扑结构存在较强依赖, 在电梯节点缺失时难以有效泛化. 相比之下, DRL-SCFO 在该场景下仍保持了较好的求解质量, 体现了更强的跨场景可迁移性. 综合以上结果可以看出, DRL-SCFO 在不同问题设置下都具有较好的泛化表现, 这使得其在面向实际 MF-PDP 及传统 PDP 场景时, 有望减少对模型重新训练的依赖, 体现了模型在不同场景下的可迁移性和可用性.

表 3 模型在不同问题规模和正态分布数据集上的泛化性能

Method	Setting	(50, 4, 4)		(100, 5, 5)		(20, 0, 1)	
		Obj.	Gap	Obj.	Gap	Obj.	Gap
AM	greedy	8.77	12.29%	14.71	14.03%	3.02	13.96%
Heter-AM	greedy	8.67	11.01%	13.81	7.67%	3.29	24.15%
MVMoE	greedy	8.73	11.78%	14.59	10.54%	3.21	21.13%
RRNCO	greedy	8.53	9.22%	14.70	13.95%	5.29	99.62%
DRL-SCFO	greedy	8.42	7.81%	13.56	5.12%	2.86	7.92%
AM	sampling-1280	8.07	3.33%	13.78	6.82%	2.72	2.64%
Heter-AM	sampling-1280	8.13	4.10%	13.59	3.49%	2.82	6.42%
MVMoE	sampling-1280	7.99	2.30%	13.57	5.19%	2.81	6.04%
RRNCO	sampling-1280	7.91	1.28%	13.70	6.20%	4.84	82.64%
DRL-SCFO(s)	sampling-1280	**7.81**	**0.00%**	**12.90**	**0.00%**	**2.65**	**0.00%**

上述实验基于合成数据分布验证了模型的泛化能力. 为进一步评估 DRL-SCFO 在分布外真实多楼层拓扑上的适用性, 本文采用 Wang 等人[29] 公开发布的真实三维多楼层建筑数据 (含 18 个主要楼层及 69 个电梯位置) 作为空间分布来源, 从中采样构造与主实验规模一致的 MF-PDP 测试实例 (每规模 10 000 个), 仅节点的空间分布来自真实场景. 各方法均使用在均匀随机实例上训练的权重直接推理.

如表 4 所示, DRL-SCFO 在三种规模上均取得了最低的目标函数值, 与对比实验结论一致. 在 (100, 5, 5) 大规模实例上, DRL-SCFO 仍优于次优方法 RRNCO, Gap 为 1.03%, 表明方法在大规模真实场景分布下仍具竞争力. 此外, MVMoE 在小规模 (20, 3, 3) 上退化明显, Gap 达 46.97%, 但在中大规模上收窄至 8.25%, 说明其混合专家结构在节点数较

表 4 真实场景数据分布下 DRL 方法性能比较

Method	(20, 3, 3)		(50, 4, 4)		(100, 5, 5)	
	Obj.	Gap	Obj.	Gap	Obj.	Gap
AM	4.81	8.22%	8.46	6.09%	13.06	2.92%
Heter-AM	4.81	8.22%	8.41	5.44%	13.22	4.23%
MVMoE	6.53	46.97%	8.63	8.25%	13.73	8.25%
RRNCO	4.53	2.11%	8.10	1.03%	12.82	1.03%
DRL-SCFO	**4.44**	**0.00%**	**7.98**	**0.00%**	**12.69**	**0.00%**

少时难以充分激活. Heter-AM 与 AM 表现接近, Gap 均在 4% 到 8% 之间, 表明异构注意力机制在分布外拓扑上未带来额外增益.

5 结 论

为应对现实世界中存在的多楼层取送货问题给现有路径优化方法所带来的挑战, 本文给出了 MF-PDP 这一问题的形式化定义, 即一类具有子区域间稀疏连接拓扑结构的新型 PDP 变体. 在此基础上, 设计了一种基于深度强化学习的稀疏跨层路径优化方法 DRL-SCFO. 该方法通过双视域空间注意力编码器 (DSAenc), 分别刻画同楼层内的密集交互关系与跨楼层间的稀疏连接结构; 同时, 引入"潜在规划-实际执行"双阶段解码器 (PDSdec), 在全局规划与局部执行之间建立协同机制, 从而有效缓解 MF-PDP 中由楼层间连接稀疏特性带来的决策挑战. 实验结果表明, DRL-SCFO 在不同 MF-PDP 实例上相较于多种主流路径规划方法取得了更优的求解性能, 并在不同数据分布及问题规模 (包括退化为传统 PDP 的情形) 下保持了稳定的求解质量, 体现出较好的泛化能力.

未来的工作将从两个层面进一步拓展本文方法. 在方法通用性方面, 将进一步完善 DSAenc 和 PDSdec 的结构设计, 使其在不依赖特定问题结构假设的情况下仍能保持较高的求解效率, 从而将 DRL-SCFO 的适用范围扩展至传统的 PDP 乃至更一般的 VRP 问题. 在问题建模方面, 当前模型假设车辆具有无限容量且电梯可即时通行, 而实际场景中车辆载重有限、电梯存在等待与运行时间. 对于有限容量约束, 可在解码阶段维护车辆剩余容量状态并据此动态收紧可行集的掩码条件, 该扩展不改变双阶段决策的基本架构, DSAenc 所建模的空间拓扑关系亦不受容量约束的影响. 对于电梯等待时间, 可将其显式纳入总路径代价, 即在计算跨楼层路径代价时叠加电梯的预估等候与运行耗时, 同时在节点特征中加入时间代价相关特征以增强编码器对动态通行代价的感知能力. 上述扩展有望在保持现有编码与解码机制核心设计的前提下, 进一步提升模型在实际多楼层运输场景中的适用性.

参考文献 (References)

[1] Savelsbergh M W P, Sol M. The general pickup and delivery problem[J]. Transportation Science, 1995, 29(1): 17-29.

[2] Hossain M. Autonomous delivery robots: A literature review[J]. IEEE Engineering Management Review, 2023, 51(4): 77-89.

[3] Custodio L, Machado R. Flexible automated warehouse: A literature review and an innovative framework[J]. The International Journal of Advanced Manufacturing Technology, 2020, 106(1): 533-558.

[4] Parragh S N, Doerner K F, Hartl R F. A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations[J]. Journal für Betriebswirtschaft, 2008, 58(2): 81-117.

[5] Kool W, Van Hoof H, Welling M. Attention, learn to solve routing problems![PP/OL]. 2018, arXiv: 1803.08475.

[6] 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.

[7] Zhou J N, Cao Z G, Wu Y X, et al. MVMoE: Multi-task vehicle routing solver with mixture-of-experts[PP/OL]. 2024, arXiv: 2405.01029.

[8] Li K W, Zhang T, Wang R, et al. Deep reinforcement learning for combinatorial optimization: Covering salesman problems[J]. IEEE Transactions on Cybernetics, 2022, 52(12): 13142-13155.

[9] Zong Z F, Zheng M, Li Y, et al. MAPDP: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(9): 9980-9988.

[10] Kong D T, Ma Y N, Cao Z G, et al. Efficient neural collaborative search for pickup and delivery problems[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 11019-11034.

[11] Son J, Zhao Z, Berto F, et al. RRNCO: Towards real-world routing with neural combinatorial optimization[C]. The 14th International Conference on Learning Representations. Rio de Janeiro: OpenReview, 2026.

[12] Bello I, Pham H, Le Q V, et al. Neural combinatorial optimization with reinforcement learning[PP/OL]. 2016, arXiv: 1611.09940.

[13] Zhou J N, Wu Y X, Cao Z G, et al. Collaboration! towards robust neural methods for routing problems[C]. Advances in Neural Information Processing Systems 37. Vancouver, Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024: 121731-121764.

[14] Chen X Y, Tian Y D. Learning to perform local rewriting for combinatorial optimization[J]. 2019: 6281-6292.

[15] Ma Y N, Li J W, Cao Z G, et al. Learning to iteratively solve routing problems with dual-aspect collaborative transformer[C]. Neural Information Processing Systems. , 2021.

[16] Ma Y N, Cao Z G, Chee Y M. Learning to search feasible and infeasible regions of routing problems with flexible neural k-opt[C]. Advances in Neural Information Processing Systems 36. New Orleans,

Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023: 49555-49578.

[17] Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]. Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2. Montreal, New York: ACM, 2015: 2692-2700.

[18] Kwon Y D, Choo J, Kim B, et al. POMO: Policy optimization with multiple optima for reinforcement learning[PP/OL]. 2020, arXiv: 2010.16011.

[19] Xin L, Song W, Cao Z G, et al. Multi-decoder attention model with embedding glimpse for solving vehicle routing problems[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(13): 12042-12049.

[20] 王万良, 陈浩立, 李国庆, 等. 基于深度强化学习的多配送中心车辆路径规划[J]. 控制与决策, 2022, 37(8): 2101-2109.
(Wang W L, Chen H L, Li G Q, et al. Deep reinforcement learning for multi-depot vehicle routing problem[J]. Control and Decision, 2022, 37(8): 2101-2109.)

[21] 周梦, 王境琦, 吴楚格, 等. 带有二维装箱约束车辆路径问题的知识驱动强化学习求解[J]. 控制与决策, 2026, 41(4): 931-943.
(Zhou M, Wang J Q, Wu C G, et al. Knowledge-driven reinforcement learning method for solving capacitated vehicle routing problem with two-dimensional loading constraints[J]. Control and Decision, 2026, 41(4): 931-943.)

[22] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[23] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 770-778.

[24] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International Conference on Machine Learning., 2015.

[25] Jang E, Gu S, Poole B. Categorical Reparametrization with Gumble-Softmax[C]. International Conference on Learning Representations (ICLR 2017). OpenReview.net, 2017.

[26] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3): 229-256.

[27] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual[CP/OL]. Beaverton: Gurobi Optimization, LLC, 2025[2025-11-21]. https://www.gurobi.com.

[28] Google LLC. OR-Tools: Google Optimization Tools[CP/OL]. Mountain View: Google LLC, 2025[2025-11-07]. https://developers.google.com/optimization.

[29] Wang Q, Veerapaneni R, Wu Y, et al. MAPF in 3D warehouses: Dataset and analysis[J]. Proceedings of the International Conference on Automated Planning and Scheduling, 2024, 34: 623-632.

作者简介

孟运 (1998−), 男, 硕士, 主要研究方向为智能优化算法与应用, E-mail: 6233114024@stu.jiangnan.edu.cn;

李超 (1990−), 男, 讲师, 博士, 主要研究方向为智能优化、演化计算、机器学习, E-mail: chaoli@jiangnan.edu.cn;

孙俊 (1971−), 男, 教授, 博士, 主要研究方向为群体智能、机器学习、具身智能, E-mail: junsun@jiangnan.edu.cn.