控制与决策
Control and Decision

面向飞机蒙皮覆盖检测的多无人机协同任务规划

朴敏楠, 李浩龙, 李海丰, 范龙飞

引用本文:
朴敏楠, 李浩龙, 李海丰, 等. 面向飞机蒙皮覆盖检测的多无人机协同任务规划[J]. 控制与决策, 2026, 41(3): 809-821.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2025.0377

面向飞机蒙皮覆盖检测的多无人机协同任务规划

朴敏楠，李浩龙，李海丰†，范龙飞

(中国民航大学 计算机科学与技术学院，天津 300300)

摘　要: 针对飞机蒙皮覆盖检测的场景下, 传统人工检测存在的作业效率低下及检测时效性约束严格等瓶颈问题, 现有研究多集中于多无人机协同作业的技术方案, 其中面向飞机蒙皮盖检测的多无人机协同任务规划(MCMP) 是描述多无人机协同检测的问题模型, 当前算法多采用启发式算法, 但其求解速度和解的质量无法满足实际要求. 为此, 将 MCMP 问题建模为带有容量约束的车辆路径规划问题 (CVRP), 提出两阶段的深度强化学习(TSDRL) 的求解模型: 第 1 阶段根据节点数量, 利用基于注意力机制的策略网络求解最优无人机数量; 第 2 阶段设计一种新的编码器-解码器结构的策略网络, 以构建每架无人机的路径. 该模型通过策略梯度训练, 能够快速求解每架无人机的高质量路径, 为了解决三维环境碰撞问题, 使用 RRT*算法优化路径以满足碰撞约束. 仿真结果表明, 所提模型在计算效率与求解质量上均优于现有的深度强化学习方法和启发式算法, 并且模型具有良好的泛化性, 可应用于不同机型.

关键词: 飞机蒙皮检测；任务规划；深度强化学习；注意力机制；策略梯度

中图分类号: TP242　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2025.0377

引用格式: 朴敏楠, 李浩龙, 李海丰, 等. 面向飞机蒙皮覆盖检测的多无人机协同任务规划 [J]. 控制与决策, 2026, 41(3): 809-821.

Multi-UAV collaborative mission planning for aircraft skin coverage detection

PIAO Min-nan，LI Hao-long，LI Hai-feng†，FAN Long-fei

(College of Computer Science and Technology，Civil Aviation University of China，Tianjin 300300，China)

Abstract: In view of the bottlenecks of traditional manual detection in the scenario of aircraft skin cover detection, such as low operation efficiency and strict detection timeliness constraints, the existing research mostly focuses on the technical solutions of multi-UAV collaborative operation, among which multi-UAV cooperative mission planning (MCMP) for aircraft skin cover detection is the problem model describing the collaborative detection of multiple UAVs, and the current algorithms mostly use heuristic algorithms. However, the speed of the solution and the quality of the solution cannot meet the actual requirements. To solve this problem, the MCMP problem is modeled as a capacitated vehicle routing problem (CVRP) with capacity constraints, and a two-stage deep reinforcement learning (TSDRL) solution model is proposed. In the first stage, the optimal number of UAVs is solved using a strategy network based on attention mechanism according to the number of nodes. In the second stage, a new encoder-decoder structure strategy network is designed to construct the path of each UAV. Trained with policy gradient methods, this model efficiently computes high-quality paths for each unmanned aerial vehicle. In order to solve the collision problem of the 3D environment, the RRT* algorithm is used to optimize the path to meet the collision constraints. Simulation results show that the proposed model is superior to the existing deep reinforcement learning methods and heuristic algorithms in terms of computational efficiency and solution quality, and the model has good generalization and can be applied to different models.

Keywords: aircraft skin inspection； mission planning； deep reinforcement learning； attention aggregation mechanism；policy gradient

收稿日期: 2025-04-11；**录用日期:** 2025-09-05.

基金项目: 国家自然科学基金项目 (U2433202, 62203450, 62373365)；航空科学基金项目 (2022Z034067004)；天津市自然科学基金多元投入青年项目 (24JCQNJC00070)；中央高校基本科研业务费专项资金项目 (3122025PT01, 3122023PT16, 3122024PT08, KJZ53420210066).

责任编委: 张雪波.

†通信作者. E-mail: hfli@cauc.edu.cn.

0 引 言

飞机蒙皮损伤 (如撞击凹坑、雷击点、裂纹等) 是影响飞行安全的重要隐患，在航线检测阶段及时识别并修复损伤至关重要. 航线检测涵盖航前、过站和航后检测, 通常面临严格的时间限制. 在实际的飞机蒙皮覆盖检测场景中, 需要先将检测任务进行分配, 但是任务会因作业场景不同而变化. 即使是同一机型, 由于周围环境 (如加油车、行李车等保障设备在机坪的位置) 的动态差异, 生成的覆盖检测任务的节点集合也各不相同, 因此可在执行检测任务前利用多无人机协同 SLAM (simultaneous localization and mapping) 技术快速构建现场环境地图, 进而基于此地图生成覆盖飞机蒙皮所需的检测节点. 若任务分配耗时过长, 则将延长整个飞机蒙皮检测任务的周期, 从而影响航班正常起飞. 航空公司需竭力缩短多架无人机覆盖检测时间, 以保障航班准点起飞. 传统的飞机蒙皮损伤检测主要依赖人工目视检查, 存在效率低、安全性差、覆盖率低、精度不足等问题. 为了克服这些局限性, 近年来, 多无人机 (multiple unmanned aerial vehicles, Multi-UAV) 技术被应用于飞机蒙皮损伤检测任务中, 通过充分发挥单机灵活性[1-4], 显著提升了航线维护检查的效率和精度.

多无人机协同覆盖任务规划 (multi-UAV cooperative mission planning, MCMP) 是实现自动化检测的核心技术, 这类技术一般应用到二维场景中的清洁与修剪任务, 以及三维环境下的模型重建、水下探测等任务[5-6]. 在飞机蒙皮检测的 MCMP 中, 主要涉及到视点生成、视点分配以及路径规划问题. 其中视点生成是基于飞机三维模型, 通过对蒙皮区域进行多种方法的分解 (如单元格分解、网格划分等), 并结合重叠率、拍摄距离、拍摄倾角等约束条件进行优化, 最终得到无人机相机位姿点, 确保检测区域的全覆盖. 而视点分配及路径规划问题则涉及将视点合理分配给每架无人机, 并确保各视点在满足碰撞约束的条件下生成飞行路径. 这些问题与车辆路径规划问题 (vehicle routing problem, VRP) 相似, VRP 问题是经典的组合优化问题 (combinatorial optimization problems, COP), 其涉及如何有效地分配一组车辆去访问多个客户点, 并在满足约束条件的情况下以最小化总行驶距离为目标去服务客户, 并且该领域核心研究问题是如何快速求解高质量解, 这恰恰与本文在多无人机覆盖检测背景下, 要求快速提供高质量解的需求相契合. 特别地, 本文将三维空间下的视点分配与路径规划问题建模为 VRP 的三维扩展模型, 并在多种约束条件下快速生成高质量路径解.

目前, 关于 VRP 不同变种问题, 已经在交通和机器人领域进行了广泛研究. 在交通运输领域, 卡车-无人机协同配送问题通常被建模为带无人机的车辆路径问题[7] 或带无人机的旅行商问题 (traveling salesman problem, TSP)[8]. 此类路径规划问题常采用混合整数线性规划 (mixed integer linear programming, MILP)[9] 进行建模, 并结合多级策略进行求解[10]. 尽管 MILP 理论上能够提供最优解, 但在处理大规模问题时计算复杂度较高[11]. 因此, 通常需要采用人工设计的启发式算法, 以在合理时间内找到次优解. 例如, 文献 [12-13] 基于 TSP 建立数学模型实现卡车与无人机的协同作业, 其中文献 [12] 的研究目标是最小化总配送时间, 而文献 [13] 则关注优化整个卡车-无人机系统的时间和能量消耗. 于彦鹏等[14] 提出一种基于进化多任务的多无人机协同路径规划算法, 将原多无人机应急配送问题作为主任务, 并将不考虑无人机续航能力和容量约束的多无人机应急配送问题当作辅助任务. 然而, 这些启发式方法通常针对特定问题类型设计, 因此其适用性和普遍性相对有限.

近年来, 基于学习的方法已成为解决 VRP 及其变体、其他 COP 的替代方案. Vinyals 等[15] 提出的神经网络模型首次将深度学习应用于 TSP 并用端到端的方式求解, 开创了深度学习在 VRP 中的应用. 随后, Kool 等[16] 引入一种基于编码器-解码器结构的 Transformer, 在多个 COP 中超越传统的启发式方法, 显著提升了解的质量和计算效率. 但是这两种方法缺乏对其他车辆状态以及 (部分) 已构建路径的考虑, 这可能导致最终生成的解的质量不高. Li 等[17] 运用深度强化学习 (deep reinforcement learning, DRL) 方法, 并结合注意力机制, 成功解决了异构容量约束 VRP, 该方法在解的质量和计算效率上超越了传统的非学习型基准方法. 此外, 王万良等[18] 提出了一种基于多智能体深度强化学习的求解模型, 利用 2-opt 局部搜索策略和采样搜索策略改进解的质量. 尽管以上研究涉及多种 VRP 的 DRL 框架, 但这些基于学习的方法会受到复杂约束等问题的影响, 导致其在解的质量上与启发式方法之间仍存在一定的差距.

综上所述, 当前 MCMP 问题的研究多集中于将其建模为 VRP 的变种问题, 并且解决其问题的算法多采用启发式方法, 虽然 DRL 在 COP 中的研究正在逐步发展, 但在航线检测背景之下仍存在以下局

限性:

1) 启发式方法依赖于专家的领域知识, 并且由于计算复杂度较高、适应性不足等问题, 导致其求解效率无法满足航线检测的要求.

2) 目前 DRL 方法中编码器在处理高维特征时会出现无法有效地提取节点特征, 并且在嵌入过程中丢失节点的语义信息, 导致节点嵌入质量不高的问题; 此外, 解码器简单地用节点嵌入均值表示图嵌入信息, 无法动态地感知路径状态的变化, 这些局限性显著影响了现有 DRL 方法在路径规划问题上的解质量和求解效率.

针对现有研究的局限性, 本文提出一种两阶段深度强化学习 (two-stage deep reinforcement learning, TSDRL), 用于解决飞行距离受限的 MCMP 问题. 本文贡献如下:

1) 针对 MCMP 问题设计一种 TSDRL 求解模型, 第 1 阶段模型根据节点数量求解出最优无人机数量, 并为第 2 阶段服务; 第 2 阶段编码器创新性引入双重批归一化以及门聚合模块, 以提高节点嵌入质量; 解码器中设计无人机选择和图聚合模块, 用于动态感知每架无人机飞行的路径状态以及节点变化情况, 为每架无人机选择合适的访问节点.

2) 通过在不同规模航线检测问题上的仿真表明, TSDRL 相比启发式算法和其他 DRL 方法, 在求解速度和解的质量上具有显著优势; 同时, 评估了 TSDRL 在不同机型的航线检测问题, 验证其具有良好的泛化性.

1　问题与模型

1.1　问题描述

在飞机蒙皮检测的 MCMP 问题中, 具备以下设定: 多 UAV 具有相同的任务起点, 每架无人机搭载相机, 在分配的节点处完成对指定飞机蒙皮区域的拍摄任务, 访问所有节点即可实现对飞机蒙皮的全覆盖. 在本文中, 节点由已有的视点生成算法[19] 给出. 在 TSDRL 模型中, 无人机对应 CVRP 模型中的车辆, 节点对应客户点, 无人机的最大飞行距离对应车辆的最大载荷, 无人机已经飞行距离与节点之间的无碰撞飞行距离相对应, 即无人机飞往节点所飞行的距离对应于 CVRP 模型中车辆行驶至客户点所消耗的装载体积. 本文的研究目标是分配一定数量的 UAV, 并通过高效求解 CVRP 实现最小化所有 UAV 的最大飞行距离. 为了便于分析和研究, 做出如下假设:

1) 任意两个节点之间的无碰撞距离均小于无人机最大飞行距离;

2) 每个节点只能被一架无人机服务;

3) 无人机之间具备相互避碰程序, 忽略任务中多架无人机避碰所增加的时间;

4) 所有无人机的最大飞行距离相同, 并以无人机的最大飞行距离作为能耗的替代约束条件;

5) 完成区域覆盖任务后, 每架无人机要返回任务起始点.

1.2　数学模型

给定$n+1$个节点 (包含 1 个任务起点和n个节点) 表示为$X=\{x^i\}_{i=0}^n$, 其中x^0表示任务起点, 节点集合定义为$X'=X\setminus\{x^0\}$. 每个节点$x^i\in\mathbb{R}^4$定义为$\{(s^i,d^i)\}$, 其中s^i包含节点x^i三维坐标, d^i表示节点的需求. 与其他 CVRP 问题不同, 这里节点x^i的需求d^i是一个动态值, 通过RRT^*算法[20]计算节点x^{i-1}到节点x^i的无碰撞距离, 有$d^i=\mathrm{RRT}^*(x^{i-1},x^i)$. 考虑到无人机动态选择情况, 根据任务进展, 选择该阶段最适合服务该节点的无人机, 因此令$V=\{v^i\}_{i=0}^{U_{\mathrm{num}}}$表示当前无人机群, 其中$v^i=\{(Q^i)\}$, Q^i为每个无人机最大飞行距离. y_{ij}^V为二元变量, 如果无人机V直接从节点x^i飞到节点x^j, 则$y_{ij}^V=1$, 否则为 0. L_{ij}^V表示无人机V飞到x^j之前剩余的飞行距离. 数学模型如下:

$$\min\max_{v\in V}\left(\sum_{i\in X}\sum_{j\in X}\mathrm{RRT}^*(x^{i-1},x^i)\times y_{ij}^v\right). \tag{1}$$

$$\text{s.t.}\ \sum_{v\in V}\sum_{j\in X}y_{ij}^v=1,\ i\in X'; \tag{2}$$

$$\sum_{i\in X}y_{ij}^v-\sum_{k\in X}y_{jk}^v=0,\ v\in V,\ j\in X'; \tag{3}$$

$$\sum_{v\in V}\sum_{j\in X}L_{ij}^v-\sum_{v\in V}\sum_{k\in X}L_{jk}^v=d^i,\ j\in X'; \tag{4}$$

$$d^jy_{ij}^v\leqslant L_{jk}^v\leqslant(Q^v-d^j)y_{ij}^v,\ v\in V,\ x,j\in X; \tag{5}$$

$$y_{ij}^v=\{0,1\},\ v\in V,\ x,j\in X; \tag{6}$$

$$L_{ij}^v\geqslant 0,d^i\geqslant 0,\ v\in V,\ x,j\in X. \tag{7}$$

式 (1) 的目标是最小化所有无人机的最大飞行距离; 约束 (2) 和 (3) 确保每个节点只被服务一次, 并且每条路线都由同一架无人机完成; 约束 (4) 保证无人机在服务节点之前和之后的飞行距离之差等于该节点的需求值; 约束 (5) 要求每架无人机的飞行距离都能满足相应节点的需求, 且不超过其最大飞行距离; 约束 (6) 定义了二元变量; 约束 (7) 规定了变量的非负性.

2 算法描述

针对所提出的TSDRL求解模型, 下面分别阐述各阶段的DRL模型, 包括马尔可夫决策过程(Markov decision process, MDP)、基于编码器-解码器结构的策略网络, 并通过策略梯度方法训练每个策略网络, 最后采用不同的动作选择策略获得高质量的解. 如图1所示, 第1阶段主要根据节点规模规划出一定数量的无人机, 并将其用于第2阶段; 在第2阶段中策略网络首先选择一架无人机, 接着为这架无人机进行任务分配, 等所有任务点分配完毕后进行路径规划.

2.1 第1阶段DRL模型

2.1.1 MDP定义

状态S_1: 状态$S_1=\{S_g,S_d\}$分为全局状态S_g和$S_d=\{U_{\rm num},E_i\}$. 其中: S_g为编码器输出的整体图特征信息, 属于静态状态; S_d会随着每个时间步t变化, $U_{\rm num}$为当前规划的无人机数量, E_i为当前无人机所走的路径点集合.

动作A_1: 第1阶段动作空间$A_1=\{a_{1,t}\}=\{x_t^i\}$表示在时间步$t$时无人机选择第$i$个节点进行服务.

状态转移τ_1: 在时间步t时, 状态转移规则τ_1根据执行的动作$A_{1,t}$将状态$S_{1,t}$转移到下一个状态$S_{1,t+1}$. 第1阶段状态$S_{d,t}$执行动作$a_{1,t}$后, 将状态转移为$S_{d,t+1}$, 即$S_{d,t+1}=\tau_1(U_{\rm num}^t,E_i^t)$. 第1阶段状态更新如下:

$$U_{\rm num}^{t+1}=\begin{cases}U_{\rm num}^t+1,\ L_{jk}^v\leqslant d^jy_{ij}^v;\\ U_{\rm num}^t,\ \text{otherwise};\end{cases}\tag{8}$$

$$E_i^{t+1}=[E_i^t,x_{t+1}^j],\ i=U_{\rm num}^t.\tag{9}$$

奖励R: 为了减少所有无人机的最大飞行距离, 奖励定义为该最大值的负值. 奖励函数可以表示为

$$R=-\max_{v\in V}\Big(\sum_{t=1}^{T}{\rm RRT}^*(A_{1,t},A_{1,t-1})\cdot y_{ij}^v\Big).$$

策略: 该阶段的目标是训练模型获得一个路径规划器, 该规划器提供的随机策略π_{θ_1}在每个时间步t根据策略网络输出的概率向量p_{θ_1}引导一架无人机在三维环境中服务节点. 最终策略输出的解$\pi=(\pi_1,\pi_2,\dots,\pi_n)$表示无人机的遍历路径, 根据链式法则可知, 随机策略π_{θ_1}输出实例s的一个完整性策略π的概率表示为

$$p(\pi|s)=\prod_{t=1}^{T}p_{\theta_1}(\pi_t|s_{t-1},\pi_{t-1}).\tag{10}$$

2.1.2 第1阶段策略网络

1) 编码器.

编码器结构如图2所示, 结构类似于文献[16]使用的编码器, 由嵌入层和N个相同结构的注意力模块构成. 编码器首先根据节点原始特征x^i计算初始节点嵌入$h_i^{(0)}=W^x\times x^i+b^x(\dim(h_i^{(0)})=128)$, 其中$W^x$和$b^x$为嵌入层网络参数; 每个注意力模块由一个多头注意力(multi-head attention, MHA)层和一个前馈(feed-forward, FF)网络层组成, 这两层均使用跳跃连接方法[21]和批归一化(batch normalization, BN)[22].

为了获得第N个注意力层的输出$h_i^{(N)}$, MHA首先通过自注意力机制[23]对$h_i^{(N-1)}$进行处理, 然后通

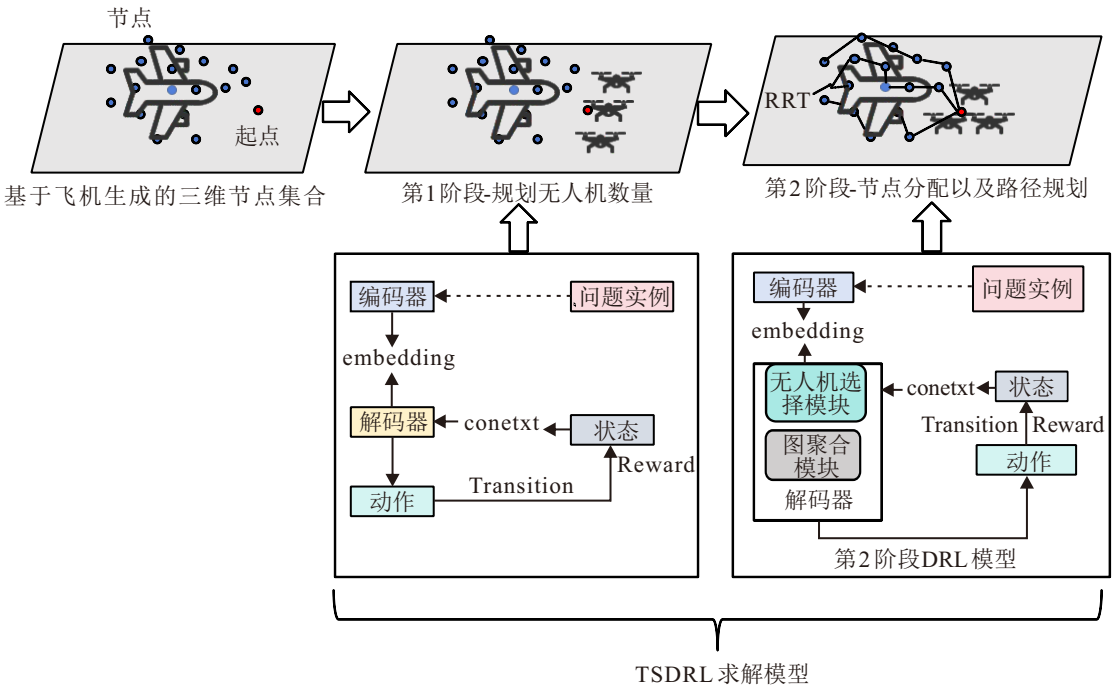


图1 两阶段算法示意图

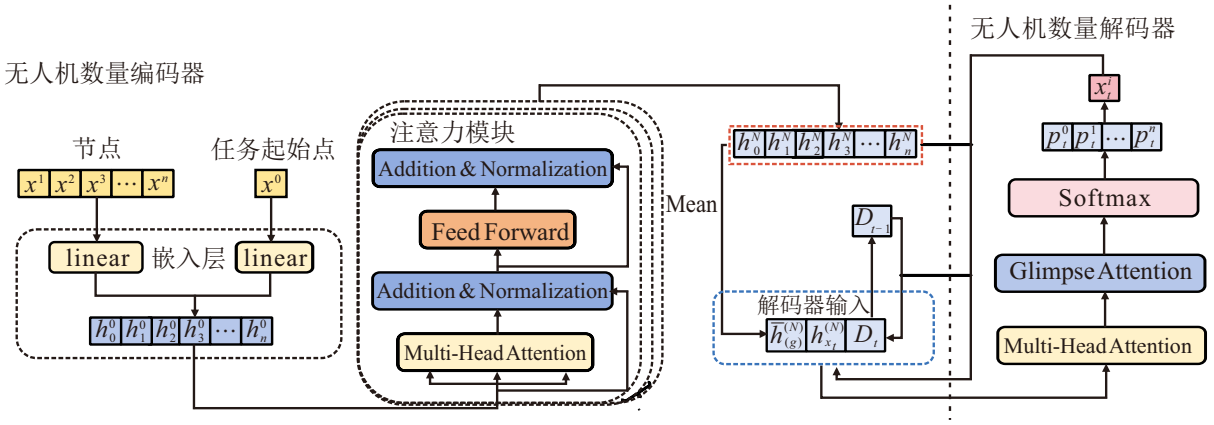


图2　第 1 阶段编码器-解码器

过跳过连接和 BN 操作得到节点嵌入 $\hat{h}_i^{(N)}$, 最后 $\hat{h}_i^{(N)}$ 被反馈到 FF 网络层, 得到最终的输出 $h_i^{(N)}$. 另外, 编码器对 $h_i^{(N)}$ 求均值来表示图嵌入 $\bar{h}_{(g)}^{(N)}$, 具体公式如下:

$$\hat{h}_i^{(N)} = \mathrm{BN}(h_i^{(N-1)} + \mathrm{MHA}_i^{(N)}(h_i^{(N-1)})), \quad (11)$$

$$h_i^{(N)} = \mathrm{BN}(\hat{h}_i^{(N)} + \mathrm{FF}^{(N)}(\hat{h}_i^{(N)})), \quad (12)$$

$$\bar{h}_{(g)}^{(N)} = \frac{1}{n+1}\sum_{i=0}^{n} h_i^{(N)}. \quad (13)$$

2) 解码器.

如图 2 所示的解码器结构, 首先根据图嵌入 $\bar{h}_{(g)}^{(N)}$、最后一个节点的嵌入 $h_{x_{t-1}^i}^{(N)}$ 以及当前无人机的剩余飞行距离 $Q_t = Q_{t-1} - \mathrm{dis}(E_i^t)$, 在时间步 t 构建解码器上下文嵌入 $h_{(ct)}$. 然后将 glimpse 注意力[23]机制应用于解码器上下文 $h_{(ct)}$, 并计算兼容性 c_t^i. 最后, 通过 softmax 运算得出在时间步长 t 服务节点的概率分布, 具体公式如下:

$$h_{(ct)} = [\bar{h}_{(g)}^{(N)}, h_{x_{t-1}^i}^{(N)}, Q_t]; \quad (14)$$

$$\tilde{h}_{(ct)} = \mathrm{glimpse}(h_{(ct)}); \quad (15)$$

$$q_{(ct)} = W^Q \tilde{h}_{(ct)},\ k_{it} = W^K h_i^{(N)},\ v_{it} = W^V h_i^{(N)}; \quad (16)$$

$$c_t^i = \begin{cases} C \cdot \tanh\left(\dfrac{q_{(ct)}^{\mathrm{T}} k_{it}}{\sqrt{d_K}}\right), & \text{满足式(1)} \sim \text{(7)}; \\ -\infty, & \text{otherwise}; \end{cases} \quad (17)$$

$$p_t(\pi_t = i | B, \pi_{1:t-1}) = \mathrm{softmax}(c_t^i). \quad (18)$$

其中: $q_{(c)}$、k_i 为用于注意力操作的 query 和 key 向量, W^Q、W^K 和 W^V 为可学习的参数. 直到所有节点都被服务, 得到序列 $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ 和最终状态无人机的数量, 其中 $N \geqslant n+1$, 包含多个任务起点, 然后进行后续训练. 重复上述步骤, 确定最优目标函数下所需无人机数量 U_{num}, 并将其传输给第 2 阶段.

2.2　第 2 阶段 DRL 模型

2.2.1　MDP 定义

状态 S_2: 第 2 阶段的状态 $S_{2,t} = (V_t, X_t)$ 由无人机状态 V_t 和节点状态 X_t 组成. 其中: $V_t = \{v_t^1, \ldots, v_t^{U_{\mathrm{num}}}\} = \{(O_t^1, G_t^1), \ldots, (O_t^{U_{\mathrm{num}}}, G_t^{U_{\mathrm{num}}})\}$, O_t^i 和 G_t^i 分别为无人机 v_t^i 在时间步 t 时所剩余飞行距离和部分行驶路径, $G_t^i = \{g_0^i, g_1^i, \ldots, g_t^i\}$ 中 g_t^i 为无人机 v_t^i 在时间步 t 服务的节点. 从任务起始点出发, 无人机初始状态为 $V_0 = \{(Q^1, \{0\}), (Q^2, \{0\}), \ldots, (Q^{U_{\mathrm{num}}}, \{0\})\}$, 其中 Q^i 为无人机 v^i 的最大飞行距离. 节点状态 X_t 具体表示为 $X_t = \{(p^0, d_t^0), \ldots, (p^n, d_t^n)\}$. 其中: p^i 为节点 x^i 位置的三维向量; d_t^i 为标量, 表示节点 x^i 在时间 t 的需求量, 一旦节点 x^i 被服务, d_t^i 便会变为 0.

动作 A_2: 第 2 阶段动作空间 $A_{2,t} = \{a_{2,t}\} = \{v_t^i, x_t^j\}$ 表示无人机 v_t^i 在时间步 t 选择节点 x_t^j 服务, 每个步骤只有一架无人机被选中.

状态转移 τ_2: 第 2 阶段状态 $S_{2,t}$ 执行动作 $a_{2,t}$ 后, 状态转移为 $S_{2,t+1} = (V_{t+1}, X_{t+1}) = \tau_2(V_t, X_t)$. 无人机状态 V_{t+1} 中的元素和节点状态更新如下:

$$O_{t+1}^k = \begin{cases} O_t^k - d_t^j, & k = i; \\ O_{t+1}^k, & \text{otherwise}. \end{cases} \quad (19)$$

$$G_{t+1}^k = \begin{cases} [G_t^k, x_t^j], & k = i; \\ [G_t^k, g_t^k], & \text{otherwise}. \end{cases} \quad (20)$$

$$x_{t+1}^l = \begin{cases} 0, & l = j; \\ x_t^l, & \text{otherwise}. \end{cases} \quad (21)$$

其中: g_t^k 为 G_t^k 的最后一个元素, 即无人机 v^k 在步骤 t 时最后服务的节点; $[\cdot, \cdot]$ 为连接操作符.

奖励 R: 奖励定义为所有无人机的最大飞行距离.

策略: 该阶段目标是学习一个随机策略 $\pi_{\theta_2}(a_{2,t} | S_{2,t})$, 其由一个具有可训练参数 θ_2 的深度神

经网络表示. 从初始状态$S_{2,0}$开始, 即一个空的解决方案, 遵循策略p_{θ_2}构建解决方案. 该过程一直持续到最终状态$S_{2,\Upsilon}$(Υ可能会超过$n+1$), 即所有的节点都被所有无人机服务. 这一过程的联合概率可以根据链式法则分解如下:

$$P(S_{2,\Upsilon}|S_{2,0}) = p_{\theta_2}(a_{2,t}|S_{2,t})P(S_{2,t+1}|S_{2,t},a_{2,t}). \tag{22}$$

式 (22) 采用确定性的状态转移, $P(S_{2,t+1}|S_{2,t},a_{2,t})=1$始终成立.

2.2.2 第 2 阶段策略网络

本文提出的二阶段编码器-解码器架构区别于传统 Transformer 的编码器-解码器结构 (如图 3 所示), 在编码器中引入了双重 BN 操作和门 (Gate) 聚合模块, 以提高节点嵌入的质量. 具体而言, 将 BN 操作合并到残差连接中, 以在 MHA 层和 FF 网络层的输入与输出之间形成恒等映, Gate 聚合模块用于 MHA 和 FF 网络层输出, 以提高节点特征的表达能力. 在解码阶段, 受到异构车辆路径问题的启发[17], 设计新的无人机选择模块, 其根据第 1 阶段得出的无人机数量$U_{\rm num}$, 能够动态感知每个无人机的状态, 并根据当前状态选择最合适的无人机. 此外, 解码器中也设计了图聚合模块用于构建未访问节点和已访问节点的图嵌入, 从而实现对上下文嵌入的动态感知, 为无人机选择合适的节点进行服务.

1) 编码器.

编码器将原始特征x^i嵌入到高维空间中, 得到每个节点嵌入. 对于第l层注意力模块, 输入$e^{(l-1)}\in\mathbb{R}^{(n+1)\times D}$表示$n+1$个节点的嵌入向量. 作为输入首先进行 BN 得到$e_{\rm BN}^{(l-1)}$, 然后$e_{\rm BN}^{(l-1)}$作为 MHA 输入得出$e_{\rm MHA}^{(l-1)}$, 有

$$e_{\rm BN}^{(l-1)} = {\rm BN}(e^{(l-1)}). \tag{23}$$

受到 GTrXL[24] 启发, 其处理强化学习任务与本文有相似之处, 因此本文将其用于状态表示学习以提高节点的嵌入质量, 并用门控输出连接替换加法操作. 输入流$e_{\rm in}$计算得到的权重因子会分配到输出流$e_{\rm out}$, 同时 MHA 的输出会通过下式进行聚合:

$$G(e_{\rm in},e_{\rm out}) = e_{\rm in}+\sigma(We_{\rm in}+b)\odot e_{\rm out}, \tag{24}$$

$$e_{\rm MHA}^{(l)} = G(e^{(l-1)},e_{\rm MHA}^{(l)}). \tag{25}$$

其中: W和b为可学习的参数, σ为 Sigmoid 函数.

在 FF 网络层中, $e_{\rm MHA}^{(l)}$作为输入先进行 BN 操作, 然后通过全连接层处理, 最终通过门聚合计算得到所有节点嵌入$e^{\rm node}$, 有

$$e^{\rm node} = G(e_{\rm MHA}^{(l)},{\rm FF}({\rm BN}(e_{\rm MHA}^{(l)}))). \tag{26}$$

2) 无人机选择模块.

在解码过程中, 根据每个无人机的状态信息和飞行路径信息, 选择一个合适的无人机并为其分配当前时间步下需要服务的节点.

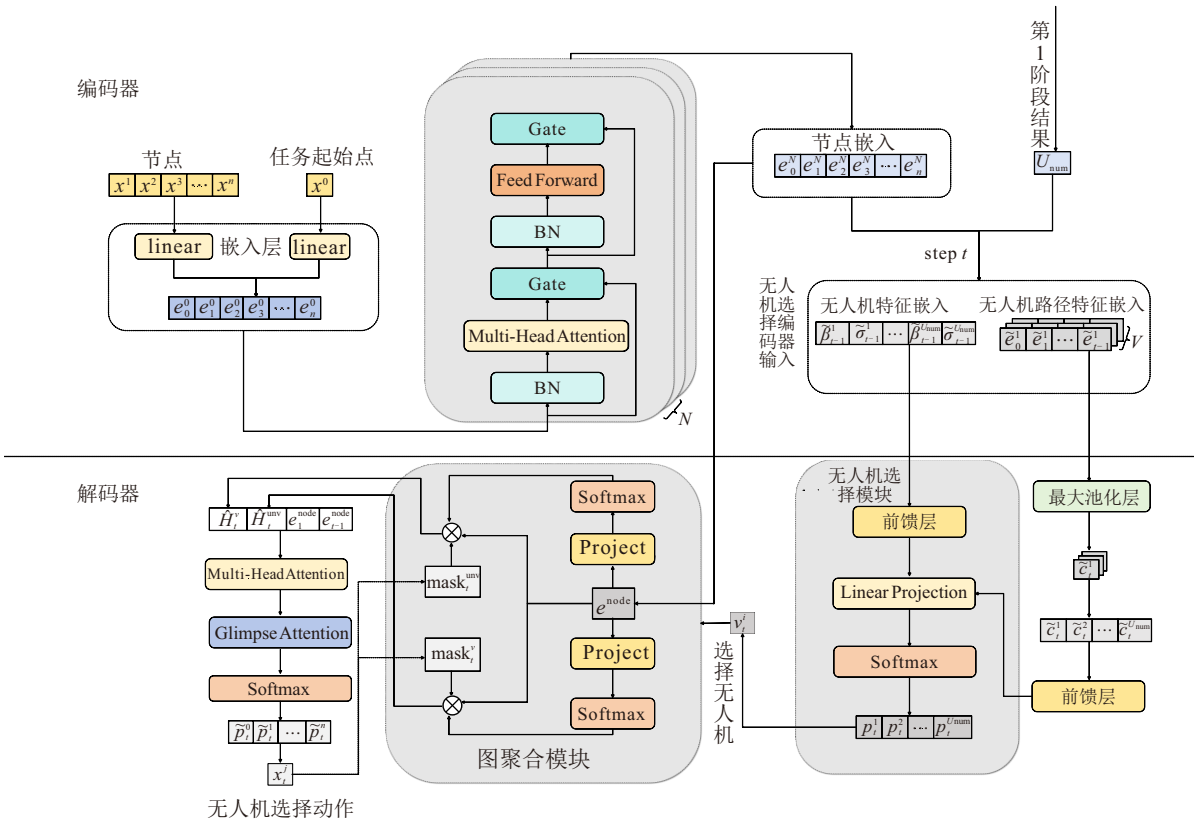


图3 第 2 阶段编码器-解码器

首先, 为每架无人机构建一个特征上下文$C_t^V=[(\beta_{t-1}^1,\sigma_{t-1}^1),\dots,(\beta_{t-1}^{U_{\text{num}}},\sigma_{t-1}^{U_{\text{num}}})]$, 以捕捉当前无人机的状态; 在时间步$t$使用线性投影和 512 维的 FF 网络层获得无人机特征嵌入H_t^V. β_{t-1}^i表示第i架无人机在时间步$t-1$所服务的最后一个节点g_{t-1}^i的三维坐标, $\sigma_{t-1}^i=(Q^i-O_{t-1}^i-\text{RRT}^*(x_t^i,x^0))$表示第$i$架无人机在时间步$t-1$可以飞回任务起始点的剩余飞行距离.

定义每架无人机v^j在时间步t的路径特征上下文$\tilde{C}_t^j=[\tilde{e}_0^j,\tilde{e}_1^j,\dots,\tilde{e}_{t-1}^j]$, 其中$\tilde{e}_{t-1}^j$为无人机$v^j$在第$t-1$时刻访问的节点嵌入信息. 将所有无人机的路径特征上下文进行最大池化再连接, 形成整个无人机群的路径上下文$\tilde{C}_t^{\mathcal{R}}=[\tilde{C}_t^1,\dots,\tilde{C}_t^{U_{\text{num}}}]$, 并对路径上下文$\tilde{C}_t^{\mathcal{R}}$进行线性投影和 512 维的 FF 网络层以获得时间步t下的路径特征嵌入$H_t^{\mathcal{R}}$. 将上述无人机特征嵌入H_t^V与路径特征嵌入$H_t^{\mathcal{R}}$连接起来, 使用带有可训练参数W_1^F和b_1的线性投影进行处理, 并通过 softmax 函数计算概率向量p_t^V, 有

$$H_t=W_1^F[H_t^V,H_t^{\mathcal{R}}]+b_1, \tag{27}$$

$$p_t^V=\text{softmax}(H_t). \tag{28}$$

3) 图聚合模块.

在基于注意力机制的 DRL 模型中[16], 上下文嵌入由图嵌入、第一个和最后一个被选择的节点嵌入 3 部分组成, 其中图嵌入通常通过所有节点嵌入平均值计算得到. 然而, 这种方法中唯一随时间变化的成分是最后一个被选择节点的嵌入, 无法动态地捕获状态转移, 限制了无人机在选择节点过程中的决策能力, 使其难以选择最优节点. 为解决这一问题, 在解码器部分设计图聚合模块. 该模块通过动态更新图嵌入, 增强上下文e_t^{context}的动态性, 使其能够更准确地反映状态转移过程中的变化, 帮助无人机选择最优的节点.

图聚合模块 (如图 4 所示) 重新定义图嵌入的结构, 将访问和未访问的节点子集建模为不同的图嵌入, 该模块在时间步t以e^{node}和掩码 (例如访问掩码mask_t^v, 其中已经访问节点标记为 1, 其余为 0) 作为输入. 在访问图嵌入构建过程中, e^{node}首先被投影以计算注意力权重, 然后与访问掩码mask_t^v和e^{node}相乘, 得到加权后的所有访问节点嵌入$\hat{e}^{\text{node}_v}$. 与 AM[16] 用节点嵌入均值作为图嵌入不同, 为了保证图的全局性特征和局部显著特征, 将$\hat{e}^{\text{node}_v}$的总和与最大值连接起来, 形成访问图嵌入$\hat{H}_t^v$, 有

$$\hat{e}^{\text{node\_}v}=\text{softmax}(W_m e^{\text{node}})\odot\text{mask}_t^v\odot e^{\text{node}}, \tag{29}$$

$$\hat{H}_t^v=\text{concat}\Big(\sum_{i=0}^{n}\hat{e}_i^{\text{node\_}v};\ \max_{i=0}^{n}\hat{e}_i^{\text{node\_}v}\Big), \tag{30}$$

其中W_m为可学习的参数.

与访问图嵌入类似, 具有参数W_m'和掩码$\text{mask}_t^{\text{unv}}$的未访问节点图嵌入以相同的机制生成. 考虑到访问节点和未访问节点具有不同语义, 本文构建两个不同的图聚合模块分别处理, 均作为解码器的上下文嵌入向量, 因此在时间步t下, 上下文嵌入向量由访问图嵌入$\hat{H}_t^v$、未访问图嵌入$\hat{H}_t^{\text{unv}}$以及无人机第 1 次和最后 1 次访问的节点嵌入组成, 表示为

$$e_t^{\text{context}}=\text{concat}(\hat{H}_t^v,\hat{H}_t^{\text{unv}},e_t^{\text{node}},e_{t-1}^{\text{node}}). \tag{31}$$

将上下文嵌入e_t^{context}和所有节点嵌入e^{node}送入 MHA 层, 并采用类似文献 [23] 的自注意力机制. 与其不同的是, 本文将e^{context}(为了简单起见, 省略时间步t) 视为f_Q, f_K和f_V不变, 对于索引为$i\in[1,n]$的每个节点, 基于$\tilde{e}_{\text{MHA}}$及节点嵌入e_i^{node}计算其注意力值μ_i, 有

$$\tilde{e}_{\text{MHA}}=\text{MHA}(f_Q=e^{\text{context}},f_K=e^{\text{node}},f_V=e^{\text{node}}). \tag{32}$$

$$\mu_i=\begin{cases}C\cdot\tanh\left(\dfrac{(W_Q\tilde{e}_{\text{MHA}})^{\text{T}}(W_K e_i^{\text{node}})}{\sqrt{d_K}}\right),\\ \quad\text{mask}_i^v=1;\\ -\infty,\ \text{otherwise.}\end{cases} \tag{33}$$

最后将μ_i输入 softmax 中, 得到对于无人机v_i所有节点的概率值$\tilde{p}_t$, 从中策略地选择动作x_t^j, 再更新访问掩码$\text{mask}_{i,t+1}^v$和未访问掩码$\text{mask}_{i,t+1}^{\text{unv}}$, 更新如下:

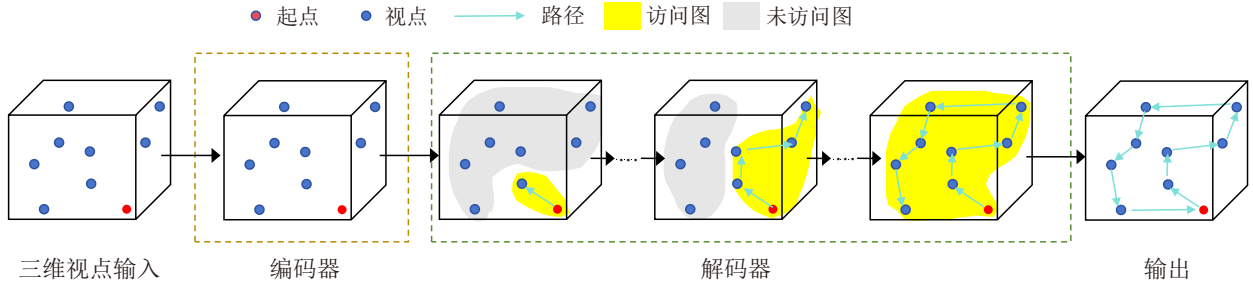


图4　图聚合示意图

$$\text{mask}_{i,t+1}^{v}=\begin{cases}1,\ i=a_{2,t};\\ \text{mask}_{i,t}^{v},\ \text{otherwise}.\end{cases}\tag{34}$$

$$\text{mask}_{i,t+1}^{\text{unv}}=1-\text{mask}_{i,t+1}^{v}.\tag{35}$$

2.3 策略网络训练方法

两阶段的强化学习均使用带有回滚基准的REINFORCE算法[25]进行训练. 这种策略梯度主要由两个网络表征: 一是策略网络$\theta=(\theta_1,\theta_2)$, 其中$\theta_1$和$\theta_2$分别代表第1阶段和第2阶段策略网络; 二是基线网络$\theta^{bl}=(\theta_1^{bl},\theta_2^{bl})$, 其中$\theta_1^{bl}$和$\theta_2^{bl}$分别代表第1阶段和第2阶段基线网络. 给定一个MCMP的实例B, 策略网络θ首先输出每一步的动作概率向量$p_\theta(\pi_t|B)$, 随后以随机采样的方式输出联合策略$\pi_t=\text{sample}(p_\theta(\pi|B))$, 而基线网络则根据其输出动作概率向量$p_{\theta^{bl}}(\pi_t|B)$以贪婪选择的方式输出联合策略$\pi_t^{bl}=\text{greedy}(p_{\theta^{bl}}(\pi|B))$. 根据蒙特卡洛算法评估策略的期望累积回报$L(\theta|B)=E_{p_\theta(B)}[R(\pi)]$, 其中$R(\pi)$为策略$\pi=\{\pi_1,\pi_2,\dots,\pi_T\}$的累积回报. 采用REINFORCE算法计算策略梯度, 使用梯度下降的方式更新策略网络参数, 即

$$\nabla_\theta L(\theta|B)=$$
$$-E_{p_\theta(B)}[(R(\pi)-R(\pi^{bl}))\nabla_\theta\log p_\theta(\pi|B)],\tag{36}$$

$$\theta=\text{Adam}(\theta,\nabla L(\theta|B)).\tag{37}$$

基线网络θ^{bl}用于评估实例B的难易程度, 从而有效减少训练过程中策略网络梯度的波动. 基线网络的更新采用回滚机制, 在每轮训练结束时, 将策略网络θ与基线网络θ^{bl}的性能进行对比. 具体而言, 通过显著性水平为$\alpha=0.05$的t检验, 若策略网络的解显著优于基线网络, 则用策略网络的参数θ更新基线网络的参数θ^{bl}. 这种更新方式能够使基线网络不断优化, 逐步接近策略网络的性能.

3 仿真分析

3.1 仿真数据及环境设置

仿真采用具有代表性的波音737-300模型, 节点数据集的生成使用文献[19]的节点生成方法, 通过该方法生成规模为30-1 (30个节点, 1个任务起始点, 无人机最大飞行距离为400 m)、50-1 (50个节点, 1个任务起始点, 无人机最大飞行距离为400 m)和90-1 (90个节点, 1个任务起始点, 无人机最大飞行距离为400 m)的不同节点数据集, 每种规模的数据各20 000套, 每套数据中包括节点数量、位置坐标以及使用RRT*计算的节点之间的无碰撞路径. 这些数据均为离线生成, 并用于两阶段模型的训练. 此外, 使用pytorch实现TSDRL算法的整体框架, 在单张GPU (1080ti, 显存64 G)上训练, 在运行环境为Intel-Core i9-CPU/2.50 GHz的win10操作系统上进行算例测试. 每种规模额外生成100套节点数据用于测试, 每套节点数据集中的任务起始点坐标均在节点三维坐标的最小值和最大值范围之外随机生成.

3.2 参数及仿真指标设置

在模型训练阶段, 每一个规模(30-1、50-1、90-1)问题下, 训练的轮次设置为100, 每一个轮次训练1 000套节点, 由于显存大小限制, 每个批次算例数设置为100, 并使用Adam优化器优化策略参数, 初始学习率设置为1×10^{-4}. 对于不同的规模问题, 分别在其对应分布下测试100套算例, 将所有测试算例中使用无人机平均数量(用#UAVs表示, 以架为单位, 结果取上界)、所有无人机的最大飞行距离(用obj表示, 以m为单位)、算法的平均求解时间(用time表示, 以s为单位)和Gap (与基准算法的差距百分比)作为模型性能评估指标. 理想情况下, 在较短的运行时间内实现的较低目标值意味着更好的解决方案质量.

3.3 与其他算法的性能对比

对于MCMP问题, 本文采用多种改进的启发式方法作为对比模型, 包括: 1) SISRs[26], 一种适用于VRP及其变体的先进启发式算法; 2) SA, 一种改进的模拟退火方法, 常用于解决CVRP及其变体问题; 3) AM[16], 该方法通过学习节点选择策略为CVRP及其变体问题提供解决方案; 4) GAT[27], 一种用残差边图注意神经网络求解CVRP及其变体问题的方法. 本文调整了所有对比模型的目标, 为确保与TSDRL算法公平比较, 将SISRs和SA中参数与对应文献保持一致, 迭代轮次随节点规模变化; 30-1规模迭代次数为1 000; 50-1规模迭代次数为10 000; 90-1规模迭代次数为100 000. 对于AM和GAT算法, 将对比其在Greedy策略和Sample1000策略下的结果, 其中Sample1000策略通过采样生成1 000个解, 并计算概率, 随后检索出最佳解. 本文所提出的TSDRL方法模型是经过离线训练得到的, TSDRL(Greedy)和TSDRL(Sample1000)分别表示TSDRL采用贪婪策略和随机采样策略. 另外, 为了显示每种方法的最优性差距, 以SISRs的求解结果为最优目标值, 计算其他算法目标函数值L与SISRs目标函数值L_{SISRs}之间的差距, 有

$$\text{Gap}=\frac{L_{\text{SISRs}}-L}{L_{\text{SISRs}}}\times100\%.\tag{38}$$

图5展示了GAT、AM和TSDRL在90-1规模下的学习曲线. 为了更清晰地展示算法学习过程, 训

练轮次设置为 100 轮, 其他仿真参数保持不变. 在训练早期 (1 ～ 30 轮次), AM 和 GAT 的学习曲线波动幅度较大; 进入 30 轮次后, 二者曲线逐渐趋于平稳. 相比之下, TSDRL 的学习曲线则较为平缓, 前 20 轮次持续呈现下降趋势, 之后开始收敛. TSDRL 不仅在收敛速度上快于 GAT 和 AM, 其获得的最终奖励值也更优. 这是由于本文针对 MCMP 问题提出的图聚合模块能够动态感知每架无人机的实时路径状态及图中节点变化, 致使策略网络高效地为每架无人机寻找到更优的路径点, 从而实现快速收敛.

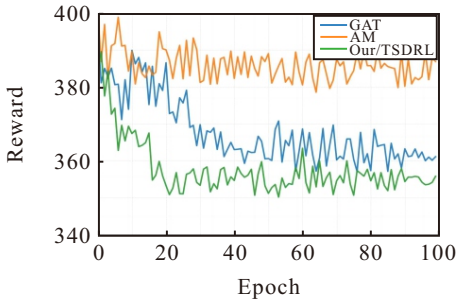


图5　90-1 规模下每种算法训练过程

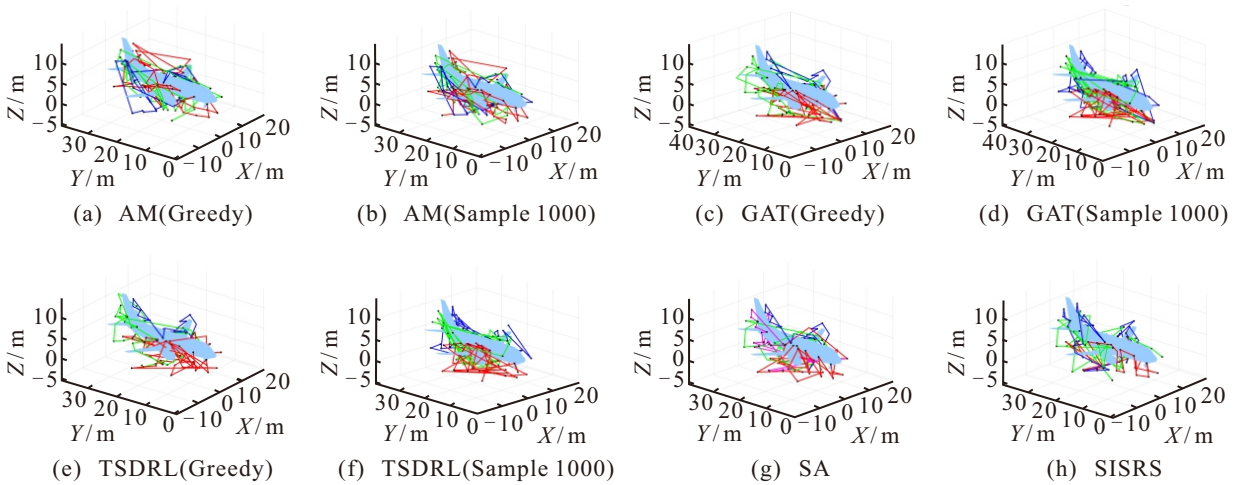
图 6 给出了这些算法在 90-1 规模下的路径规划示意图 (含RRT*), 图中每种颜色路线代表每架无人机路径规划结果. 本文通过设计双重 BN、Gate 聚合以及图聚合模块, 有效考虑了访问节点与未访问节点的动态权重分配, 从而显著提升了无人机动作选择的最优性. 表 1 ～ 表 3 列出了这些算法在不同规模下的性能, 由此可见: 本文提出的 TSDRL 算法 (无论是 Greedy 还是 Sample 1000) 在不同规模下的解的质量和求解效率均优于 AM 和 GAT 算法, 并且在 30-1 规模下这种优势显著. 虽然 SA 算法使用较多的无人机, 其解的质量优于 TSDRL (Greedy) 和 AM 算法, 但仍不及 TSDRL (Sample 1 000), 且在求解效率上也不如 TSDRL. SISRs 算法在不同规模下始终保持最优解的质量, 尽管 TSDRL 在解的质量上略逊于 SISRs, 但在求解效率上更具优势, 这种差距会随着节点规模的增加而减少. 综上所述, TSDRL 算法在解的质量和计算效率方面相较于 AM 和 GAT 算法表现出显著优势. 特别是随着问题规模的增大, SISRs 和 SA 的计算时间几乎呈指数级增长, 而 TSDRL 的计算时间则呈线性增长, 因此在航线检测这种背景下, TSDRL 算法具有较高的求解效率和良好的解质量, 其能够更快速地适应复杂的航线检测问题中.

3.4　消融实验

为了评估双重归一化和门聚合模块及图聚合模块在 TSDRL 框架中的贡献, 设计了严格的消融实验. 实验设置 4 个对比组: 1) TSDRL-N1 保留双重归一化与门聚合模块但移除图聚合模块; 2) TSDRL-N2 保留图聚合模块但移除双重归一化与门聚合模块; 3) TSDRL-N3 同时移除上述 3 个关键模块; 4) 完整 TSDRL 作为对照组.

如图 7 所示, 在 90-1 规模下, 4 个对比组的训练曲线呈现显著差异. TSDRL-N1 展现出最快的收敛速度, 且训练过程最为平稳. 相比之下, TSDRL-N3 的收敛速度较慢. TSDRL-N2 在训练初期呈现下降趋势, 且收敛后的奖励值低于其他消融组. 综合而言, TSDRL 融合了 TSDRL-N1 与 TSDRL-N2 的优势, 不仅训练过程更为稳定, 而且更快收敛至最优解.

表 4 ～ 表 6 的详细对比数据进一步表明: 在模块协同方面, 双重归一化与门聚合模块 (TSDRL-N1 组) 和图聚合模块 (TSDRL-N2 组) 在 30-1 和 50-1 规模下均表现出显著的性能互补性. 在 30-1 规模中, TSDRL-N1 (Sample) 与 TSDRL-N2 (Sample) 差距分别为 32.93% 和 27.19%, 而完整 TSDRL 将差距降至 18.48%, 这种现象在 90-1 规模中更为明显, 表明随着问题规模增大, 两个模块的协同效果表现更好.



(a) AM(Greedy)　(b) AM(Sample 1000)　(c) GAT(Greedy)　(d) GAT(Sample 1000)

(e) TSDRL(Greedy)　(f) TSDRL(Sample 1000)　(g) SA　(h) SISRS

图6　90-1 规模下每种算法路径规划示意图

表1 30-1 规模下不同算法求解结果对比

算法	obj./m	time/s	#UAVs	Gap/%
SISRS	209.45	334.68	1	—
SA	259.52	133.45	2	14.36
AM (Greedy)	362.83	0.59	1	73.22
AM (Sample 1000)	285.46	5.80	1	36.29
GAT (Greedy)	302.74	0.96	1	44.54
GAT (Sample 1000)	248.16	20.15	1	28.37
TSDRL (Greedy)	296.43	0.57	1	41.52
TSDRL (Sample 1000)	248.16	5.80	1	18.45

表2 50-1 规模下不同算法求解结果对比

算法	obj./m	time/s	#UAVs	Gap/%
SISRs	257.12	589.46	2	—
SA	348.62	317.23	3	35.58
AM (Greedy)	371.96	0.62	2	44.66
AM (Sample 1000)	350.20	6.60	2	36.20
GAT (Greedy)	367.87	0.88	2	43.07
GAT (Sample 1000)	344.94	24.48	2	34.15
TSDRL (Greedy)	354.21	0.59	2	38.20
TSDRL (Sample 1000)	334.59	6.40	2	30.12

表3 90-1 规模下不同算法求解结果对比

算法	obj./m	time/s	#UAVs	Gap/%
SISRs	329.16	1 108.47	4	—
SA	366.55	784.62	5	11.3
AM (Greedy)	389.45	0.66	4	18.31
AM (Sample 1000)	367.25	8.80	4	11.57
GAT (Greedy)	370.86	1.33	4	12.66
GAT (Sample 1000)	364.83	32.15	4	10.83
TSDRL (Greedy)	368.32	0.61	4	11.89
TSDRL (Sample 1000)	352.63	8.60	4	7.14

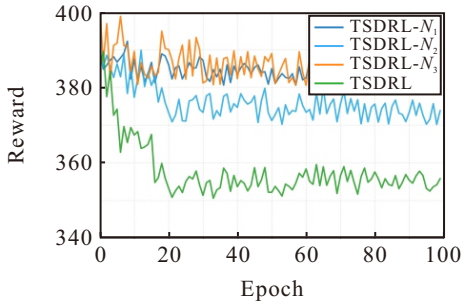


图7 90-1 规模下消融实验训练过程

在模块的规模适应性方面, 图聚合模块在小规模场景 (30-1) 中展现出更突出的贡献 (TSDRL-N2 比 TSDRL-N1 提升 5.74 个百分点), 这表明图聚合模块更擅长处理较小规模局部特征交互, 而双重归一化与门聚合模块在大规模场景 (90-1) 中表现出更好的可扩展性.

表4 30-1 规模下求解结果对比

算法	obj./m	time/s	#UAVs	Gap/%
TSDRL-N1 (Greedy)	320.55	0.52	1	53.04
TSDRL-N1 (Sample 1000)	278.43	5.40	1	32.93
TSDRL-N2 (Greedy)	304.61	0.55	1	45.43
TSDRL-N2 (Sample 1000)	266.40	5.28	1	27.19
TSDRL-N3 (Greedy)	362.83	0.59	1	73.23
TSDRL-N3 (Sample 1000)	285.46	5.80	1	36.29
TSDRL (Greedy)	296.43	0.57	1	41.53
TSDRL (Sample 1000)	248.16	5.80	1	18.48

表5 50-1 规模下求解结果对比

算法	obj./m	time/s	#UAVs	Gap/%
TSDRL-N1 (Greedy)	370.11	0.55	2	43.94
TSDRL-N1 (Sample 1000)	345.25	5.23	2	34.28
TSDRL-N2 (Greedy)	360.10	0.58	2	40.05
TSDRL-N2 (Sample 1000)	340.55	5.92	2	32.45
TSDRL-N3 (Greedy)	371.96	0.62	2	44.66
TSDRL-N3 (Sample 1000)	350.20	6.60	2	36.20
TSDRL (Greedy)	354.21	0.59	2	37.76
TSDRL (Sample 1000)	334.59	6.40	2	30.13

表6 90-1 规模下求解结果对比

算法	obj./m	time/s	#UAVs	Gap/%
TSDRL-N1 (Greedy)	381.43	0.59	4	15.88
TSDRL-N1 (Sample 1000)	362.71	7.82	4	10.19
TSDRL-N2 (Greedy)	370.98	0.63	4	12.71
TSDRL-N2 (Sample 1000)	358.64	7.82	4	8.96
TSDRL-N3 (Greedy)	389.45	0.66	4	18.32
TSDRL-N3 (Sample 1000)	367.25	8.80	4	11.57
TSDRL (Greedy)	368.32	0.61	4	11.90
TSDRL (Sample 1000)	352.63	8.60	4	7.13

3.5 计算开销分析及关键参数影响

3.5.1 计算开销分析

1) 第 1 阶编码器-解码器.

编码器: 采用 6 层 Transformer 结构, 每层包含多头注意力 MHA 和前馈网络 FFN. MHA 计算复杂度为 $O(N^2 \cdot D)$ (N为节点数, $D=128$), FFN 为 $O(N \cdot D^2)$, 总编码器复杂度为 $6(O(N^2 \cdot D)+O(N \cdot D^2))$.

解码器: 每个时间步t下, 上下文构建 (拼接图嵌入$h_g^{(N)}$、上一节点嵌入$h_{t-1}^{(N)}$、剩余距离Q_t) 时间复杂度为$O(1)$, Glimpse 注意力 (计算兼容c_i^t, 需遍历所有节点) 复杂度为$O(N)$, 注意力得分复杂度为

$O(N)$, Softmax 概率分布计算复杂度为$O(N)$, 因此单步复杂度为$O(N)$. 总时间步数需服务N个节点(不含起点), 解码器总复杂度为$O(N^2)$.

2) 第 2 阶段编码器-解码器.

编码器: 采用 6 层 Transformer 结构, 每层包含多头注意力 (MHA) 和前馈网络 (FFN), MHA 计算复杂度为$O(N^2 \cdot D)$(N为节点数, $D = 128$), FFN 为$O(N \cdot D^2)$, 双重批归一化和门控聚合增加$O(N \cdot D)$开销, 总编码器复杂度为$6(O(N^2 \cdot D) + O(N \cdot D^2) + O(N \cdot D))$.

解码器: 动态图聚合模块需计算未访问/已访问子图注意力 (各$O(N \cdot D^2)$), 无人机选择模块的路径特征处理引入$O(U_{\text{num}} \cdot D^2)$开销 ($U_{\text{num}}$为无人机数量). 单步解码复杂度为$O(N \cdot D^2 + U_{\text{num}} \cdot D^2)$, 完整解码的复杂度为$O(N^2 \cdot D^2)$.

3.5.2 关键参数

本文重点考虑模型结构中的核心参数, 其中包括编码器注意力头数、节点嵌入维度、编码器层数, 这些参数直接决定了模型的表示能力、复杂度和计算量. 其中: M(注意力头数) 主要影响模型并行捕捉不同特征或关系的能力; dim(节点嵌入维度) 定义了模型中特征表示的基本宽度, 直接影响模型学习复杂模式的能力和参数规模; N(编码器层数) 决定了模型进行特征抽象和转换的深度, 对捕捉长程依赖和层次化信息至关重要. 相比之下, 学习率属于训练优化范畴的超参数, 其最佳值通常与模型结构、优化器选择及数据特性强相关. 类似地, 前馈神经网络 (FFN) 的层数虽然重要, 但在 Transformer 架构中通常被视为次级设计参数或与dim关联, 其独立变化对模型核心能力的影响通常不如M、dim、N显著. 因此, 本文将M、dim、N作为核心变量进行分析. 以 90-1 规模为例, 两阶段模型参数保持一致, 通过改变 3 种模型参数来观察最终解, 如表 7 ～ 表 9 所示.

由表 7 ～ 表 9 可见, 在注意力头数方面, 当$M = 8$时, Greedy 和 Sample1000 策略的目标函数均值达到最低值, 但当M扩大至 16 时, 求解结果反而上升, 表明头数过多会降低模型表征能力. 在节点嵌入维度方面, 128 维时性能最优 (Greedy : 368.32 m, Sample :352.63 m), 当维度提升至 256 和 512 时, 求解结果不减反增, 同时求解时间递增, 表明更高维度未带来表征增益反而可能引入冗余噪声. 在编码器层数方面, 6 层结构 ($N = 6$) 显著优于 4 层 (Greedy 结果差距 4.77 m), 但增至 8 层时性能小幅下降 (Greedy : 369.73 m) 且计算耗时明显增加, 表明层数过深可能导致优化困难. 因此最优参数组合 ($M = 8$, $\text{dim} = 128$, $N = 6$) 在解质量与效率间取得最佳平衡.

表7　90-1 规模下不同注意力头数求解结果对比

关键参数	obj./m	time/s	#UAVs	Gap/%
TSDRL(Greedy, $M = 4$)	379.40	0.55	4	15.26
TSDRL(Sample 1000, $M = 4$)	369.12	7.2	4	12.14
TSDRL(Greedy, $M = 8$)	368.32	0.61	4	11.90
TSDRL(Sample 1000, $M = 8$)	352.63	8.60	4	7.13
TSDRL(Greedy, $M = 16$)	377.43	0.72	4	14.66
TSDRL(Sample 1000, $M = 16$)	367.29	9.4	4	11.58

表8　90-1 规模下不同节点嵌入维度求解结果对比

关键参数	obj./m	time/s	#UAVs	Gap/%
TSDRL(Greedy, $\text{dim} = 128$)	369.22	0.66	4	11.90
TSDRL(Sample 1000, $\text{dim} = 128$)	352.63	8.30s	4	7.13
TSDRL(Greedy, $\text{dim} = 256$)	370.47	0.61	4	12.55
TSDRL(Sample 1000, $\text{dim} = 256$)	360.12	8.60	4	9.41
TSDRL(Greedy, $\text{dim} = 512$)	375.90	0.74	4	14.20
TSDRL(Sample 1000, $\text{dim} = 512$)	367.19	9.35	4	11.55

表9　90-1 规模下不同编码层数求解结果对比

关键参数	obj./m	time/s	#UAVs	Gap/%
TSDRL(Greedy, $N = 4$)	373.09	0.53	4	13.35
TSDRL(Sample 1000, $N = 4$)	368.43	7.89	4	11.93
TSDRL(Greedy, $N = 6$)	368.32	0.61	4	11.90
TSDRL(Sample 1000, $N = 6$)	352.63	8.60	4	7.13
TSDRL(Greedy, $N = 8$)	369.73	0.79	4	12.33
TSDRL(Sample 1000, $N = 8$)	365.87	10.41	4	11.16

3.6 跨机型泛化性检验

为了评估所提出 TSDRL 模型的泛化能力, 本文通过调整问题场景中的节点数量, 测试不同规模的实例. 在实际应用中, 不同类型的飞机在采用相同节点生成方法时, 其节点数量可能存在差异, 因此将该模型扩展至不同类型飞机具有重要的现实意义. 基于波音 737-100 生成 3 种规模的数据集 (20-1、40-1、80-1), 并将 TSDRL 模型在波音 737-300 的 90-1 规模问题中学习到的策略应用于新生成的实例, 将 TSDRL 与 AM、GAT 等基准方法进行对比, 以评估模型的泛化性. 对比结果如图 8 所示, 在求解目标函数方面, TSDRL (无论是 Greedy 还是 Sample 1000) 均优于 AM 和 GAT, 因此 TSDRL (Sample 1000) 具有更好的泛化性能力. 随着问题规模的增大, 4 种方法的平均最大飞行距离呈现增长趋势. 但是由于无

人机最大飞行距离的限制, 4 种算法之间差距逐渐趋于平稳. 随着问题规模增大, TSDRL (Sample 1000) 模型优势更加明显. 综上所述, TSDRL 模型具有良好的泛化能力.

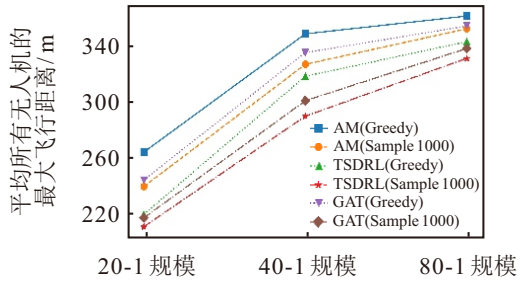


图8 模型泛化求解性能对比

4 结 论

针对飞机需要快速进行航线检测的场景, 本文提出了一种基于深度强化学习的求解模型. 该模型第 1 阶段根据节点数量求解无人机数量, 第 2 阶段将编码器引入双重批归一化和门聚合以提高节点嵌入质量, 解码器设计无人机选择和图聚合模块, 动态地反应状态转换, 为无人机选择最优动作. 经过离线训练的 TSDRL 模型能够快速求解 MCMP. 通过大量算例的对比仿真验证了 TSDRL 模型在求解速度上始终保持最优, 并且在目标函数值的优化上, 相较于 AM 和 GAT 算法表现出更优的解质量. 后续研究将重点探讨模型在多起点问题以及在更大规模问题中的求解能力, 并设计更为高效的求解模型.

参考文献 (References)

[1] Liu Y P, Dong J X, Li Y D, et al. A UAV-based aircraft surface defect inspection system via external constraints and deep learning[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 5016113.

[2] Yasuda Y D V, Cappabianco F A M, Martins L E G, et al. Aircraft visual inspection: A systematic literature review[J]. Computers in Industry, 2022, 141: 103695.

[3] Saha A, Kumar L, Sortee S, et al. An autonomous aircraft inspection system using collaborative unmanned aerial vehicles[C]. 2023 IEEE Aerospace Conference. Big Sky, 2023: 1-10.

[4] 戴佳佳, 龚小溪, 汪俊. 面向飞机外表面检测任务的无人机覆盖路径规划方法[J]. 机械工程学报, 2023, 59(16): 243-253.
(Dai J J, Gong X X, Wang J. UAV coverage path planning method for aircraft exterior surface detection task[J]. Journal of Mechanical Engineering, 2023, 59(16): 243-253.)

[5] Oh J S, Park J B, Choi Y H. Complete coverage navigation of clean robot based on triangular cell map[C]. 2001 IEEE International Symposium on Industrial Electronics Proceedings. Pusan, 2001: 2089-2093.

[6] Zhang Y X, Wang Q, Shen Y, et al. An online path planning algorithm for autonomous marine geomorphological surveys based on AUV[J]. Engineering Applications of Artificial Intelligence, 2023, 118: 105548.

[7] Wang Z, Sheu J B. Vehicle routing problem with drones[J]. Transportation Research Part B: Methodological, 2019, 122: 350-364.

[8] Tang Z Y, van Hoeve W J, Shaw P. A study on the traveling salesman problem with a drone[C]. Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Cham: Springer International Publishing, 2019: 557-564.

[9] Sundar K, Venkatachalam S, Rathinam S. Formulations and algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem[C]. 2016 American Control Conference. Boston, 2016: 6489-6494.

[10] Liu Y, Liu Z, Shi J M, et al. Two-echelon routing problem for parcel delivery by cooperated truck and drone[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(12): 7450-7465.

[11] Cattaruzza D, Absi N, Feillet D. Vehicle routing problems with multiple trips[J]. Annals of Operations Research, 2018, 271(1): 127-159.

[12] Wang C, Lan H J. An expressway based TSP model for vehicle delivery service coordinated with truck + UAV[C]. 2019 IEEE International Conference on Systems, Man and Cybernetics. Bari, 2019: 307-311.

[13] Remer B, Malikopoulos A A. The multi-objective dynamic traveling salesman problem: Last mile delivery with unmanned aerial vehicles assistance[C]. 2019 American Control Conference. Philadelphia, 2019: 5304-5309.

[14] 于彦鹏, 余墨多, 汤奇荣, 等. 面向城市应急物资配送的多无人机协同路径规划算法[J]. 控制与决策, 2025, 40(4): 1098-1106.
(Yu Y P, Yu M D, Tang Q R, et al. Multi-UAV collaborative path planning algorithm for urban emergency material distribution[J]. Control and Decision, 2025, 40(4): 1098-1106.)

[15] Vinyals O, Fortunato M, Jaitly N. Pointer networks[J]. Advances in Neural Information Processing Systems, 2015, 28: 2692-2700.

[16] Kool W, van Hoof H, Welling M. Attention, learn to solverouting problems![J/OL]. 2018, arXiv: 1803.08475.

[17] 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, 2022, 52(12): 13572-13585.

[18] 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. Vehicle routing planning for multiple distribution centers based on deep reinforcement learning[J]. Control and Decision, 2022,

37(8): 2101-2109.)

[19] 朴敏楠, 周雨晗, 李海丰, 等. 摄影测量约束下无人机分层覆盖路径规划[J]. 计算机工程与应用, https://link.cnki.net/urlid/11.2127.tp.20241223.1133.012.
(Piao M N, Zhou Y H, Li H F, et al. Hierarchical coverage path planning of UAV under photogrammetry constraints[J]. Computer Engineering and Applications, https://link.cnki.net/urlid/11.2127.tp.20241223.1133.012.)

[20] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7): 846-894.

[21] 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.

[22] Ioffe S. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J/OL]. 2015, arXiv: 1502.03167.

[23] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Proceedings of the 31st Conference on Neural Information Processing Systems. Los Angeles: Curran Associates, 2017: 5998-6008.

[24] Parisotto E, Song F, Rae J, et al. Stabilizing transformers for reinforcement learning[C]. Proceedings of the 37th International Conference on Machine Learning. Virtual: PMLR, 2020: 7487-7498.

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

[26] Christiaens J, Vanden Berghe G. Slack induction by string removals for vehicle routing problems[J]. Transportation Science, 2020, 54(2): 417-433.

[27] Lei K, Guo P, Wang Y, et al. Solve routing problems with a residual edge-graph attention neural network[J]. Neurocomputing, 2022, 508: 79-98.

作者简介

朴敏楠 (1993−), 女, 副教授, 博士, 硕士生导师, 主要研究方向为机器人运动规划与控制, E-mail: mnpiao@cauc.edu.cn;

李浩龙 (2000−), 男, 硕士生, 主要研究方向为机器人强化学习, E-mail: 2023052063@cauc.edu.cn;

李海丰 (1984−), 男, 教授, 博士, 主要研究方向为机器人环境感知、计算机视觉, E-mail: hfli@cauc.edu.cn;

范龙飞 (1989−), 男, 实验师, 硕士生, 主要研究方向为机器学习, E-mail: lffan@cauc.edu.cn.