控制与决策
Control and Decision

基于Transformer-DRL的机坪特种车群调度策略研究

陈维兴, 李晨辉, 李业波

引用本文:
陈维兴, 李晨辉, 李业波. 基于Transformer-DRL的机坪特种车群调度策略研究[J]. 控制与决策, 2025, 40(6): 1939-1949.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.0918

基于 Transformer-DRL 的机坪特种车群调度策略研究

陈维兴[1†]，李晨辉[1]，李业波[2]

(1. 中国民航大学 电子信息与自动化学院，天津 300300；
2. 中国民用航空华北地区空中交通管理局河北分局，石家庄 050802)

摘　要: 针对机坪环境下多种类地面服务车辆的协同调度这一复杂的优化任务, 提出一种结合 Transformer 架构的深度强化学习算法. 首先, 依据航班地面服务流程的不同优先级, 将整个地面服务任务进行分解, 进而将原本复杂的多类型车辆调度问题转化为有先后顺序的单类型车辆调度问题; 接着, 利用 Transformer 架构对航班和车辆的特征进行自动提取, 通过解码器按序列逐步求解任务调度, 结合贪婪算法和蒙特卡洛模拟算法分别生成初步调度策略, 并将这些策略应用于每个子问题的求解过程中; 在此基础上, 利用深度强化学习算法对整个模型进行训练, 通过智能体与环境的交互来不断优化调度策略; 此外, 为了提升模型的鲁棒性和应对复杂情况的能力, 通过扩充真实数据集进行模型训练. 大量的实验结果证明, 基于 Transformer 架构的深度强化学习方法能够有效避免不同种类车辆之间的相互干扰, 并很好地应对真实环境下的航班调度需求.

关键词: 机坪特种车辆；多车型动态调度；深度强化学习；Transformer 架构

中图分类号: V351^{+}.3；TP181　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.0918

引用格式: 陈维兴, 李晨辉, 李业波. 基于 Transformer-DRL 的机坪特种车群调度策略研究 [J]. 控制与决策, 2025, 40(6): 1939-1949.

Research on scheduling strategy of special vehicle cluster on apron based on Transformer-DRL

CHEN Wei-xing[1†]，*LI Chen-hui*[1]，*LI Ye-bo*[2]

(1. College of Electronic Information and Automation， Civil Aviation University of China， Tianjin 300300，China； 2. HeBei Sub Bureau of North China Regional Air Traffic Management Bureau. CAAC， Shijiazhuang 050802，China)

Abstract: Aiming at the complex optimization task of collaborative scheduling of multiple types of ground service vehicles in the ramp environment, this paper proposes a deep reinforcement learning algorithm integrated with the Transformer architecture. First, the entire ground service task is decomposed based on the varying priorities of the flight ground service process, transforming the complex multi-type vehicle scheduling problem into a sequential single-type vehicle scheduling problem. The Transformer architecture is then employed to automatically extract the features of flights and vehicles, and task scheduling is solved step by step through the decoder. Preliminary scheduling strategies are generated by combining greedy and Monte Carlo simulation algorithms, which are applied to each sub-problem. On this basis, a deep reinforcement learning algorithm is used to train the entire model, continuously optimizing the scheduling strategy through the interaction between the agent and the environment. Further, to enhance the model's robustness and ability to handle complex situations, the model is trained by expanding the real dataset. Extensive experiments demonstrate that the deep reinforcement learning approach based on the Transformer architecture effectively prevents mutual interference among different vehicle types and can meet the flight scheduling requirements in real-world environments.

Keywords: apron special vehicles； multi-vehicle dynamic scheduling； deep reinforcement learning； Transformer architecture

收稿日期: 2024-07-30；**录用日期:** 2024-11-10.
基金项目: 天津市教委自然科学科研基金项目 (2018KJ237).
责任编委: 唐加福.
[†]通信作者. E-mail: cw007x130@vip.163.com.
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.

0 引 言

随着民航运输业的蓬勃发展, 机场的旅客吞吐量以及航班起降架次不断增加, 虽然给航空公司与机场带来了巨大的经济效益, 但也对相关单位整合资源的能力提出了更高的要求. 机场地面保障服务是一项复杂的系统工程[1], 其中机场特种车辆调度问题是研究的焦点之一. 对于这类问题, 已经有很多专家学者用精确的分支定界算法或元启发式算法进行求解, 并且对此类算法进行改进并应用在牵引车[2]、摆渡车[3] 以及行李车调度等问题中[4]. 然而, 这类方法在面对大规模, 尤其是多种车辆协同调度问题时, 通常需要消耗大量时间, 且无法处理机场实际运行中灵活多变的航班起降时间.

针对上述问题, 文献 [5] 提出了利用随机参数约束的方法对航班时间进行处理, 在航班到达时间的基础上有概率地叠加时间偏差来模拟真实的机场运行场景, 但是没有考虑完整地面服务流程的问题且依然无法解决运行时间过长的问题. 文献 [6] 利用真实的航班时间, 结合调度问题的马尔可夫特性进行求解, 利用深度Q网络 (DQN), 根据不同时刻的航班动态选择不同的员工分配方法, 实现了综合目标值的动态优化. 但其仍以人工经验为基础, 无法全面地提取航班和工作人员的动态特征以及评估不同指标的价值. 文献 [7] 基于深度强化学习算法开发了一种专门针对机场地面车辆协同调度的端到端框架, 包含了马尔可夫决策过程 (MDP) 中状态空间的建立以及选择动作的策略, 改善了对人工经验的依赖, 并解决了多类型车辆的协同服务问题. 然而, 其将构建的动作选择策略应用于所有种类的车辆调度中, 忽视了不同类型服务约束条件, 且解码器中车群整体嵌入的方法造成了不同种类车辆特征融合, 在构造解集时对相互学习造成干扰.

综上, 本文引入 Transformer 框架经典的编码器 —— 解码器结构, 依靠深度强化学习算法对模型参数进行训练, 在此基础上对其进行改进, 引入完全相同的两个策略网络[8], 在训练中打乱不同车辆融合在一起的特征嵌入, 且在计算不同优先级服务时对车辆进行有选择地嵌入, 排除不同优先级车辆之间的相互干扰. 具体内容如下:

针对多类型特种车辆调度问题, 首先基于航班服务流程的不同优先级对复杂问题进行分解, 将问题分解为多个子问题, 并针对各自的特征构建混合整数规划模型进行求解; 随后, 利用所设计的策略网络对这些子问题进行求解优化, 综合求解结果以获得完整的解集; 建立 MDP 模型, 重点构建动作选择策略网络, 利用端到端模型选择下一时刻待服务的航班, 并基于此对深度强化学习模型进行训练; 最后模拟天津滨海国际机场的真实数据对训练好的模型进行验证.

1 多类型特种车辆调度

1.1 问题描述与详解

传统意义上, 地面保障车辆主要根据航班计划表的预计到达时间与起飞时间, 同时综合地面保障车辆情况来确定车辆的路径与服务时间, 以满足航班保障需要[9]. 因此, 本质上这一调度过程可以看作是对多类型地面服务车辆的保障流程进行分解, 并通过数学建模将其整合为一个混合整数规划模型.

在机场实际运营中, 航班每一种服务类型都由一组同种类型车辆组成的车队完成, 因此该问题可以看作多车队带时间窗的车辆路径问题. 具体而言, 以最小化总体车辆的总路径为总目标, 将每一架飞机视为节点, 去往有服务需求飞机的路径视为一条边, 而飞机的保障服务包含不同的种类, 每一种服务视为一种独立操作, 所有的服务组成完整的操作集, 并由不同类型的车群执行.

此外, 在实际的机坪环境中, 各类飞机保障服务之间存在不同的优先顺序要求, 因此需要依据服务流程确定服务之间的优先级关系, 具体要求如图 1 所示.

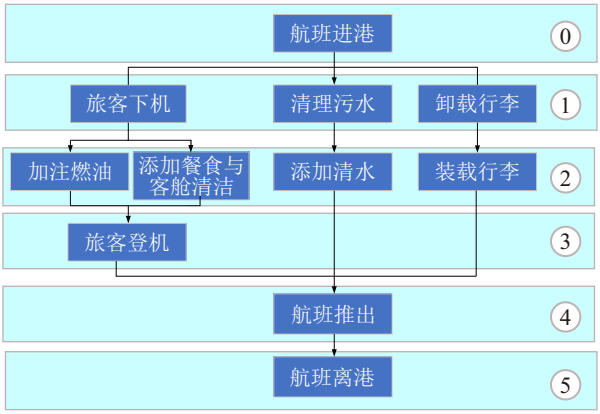

图1 航班地面服务流程

通过将复杂的地面服务分解为各类型车辆各自服务的子问题, 把多类型车辆协同调度转化为多个单类型车辆调度的集合, 再按照保障服务流程, 根据不同的优先级对各子问题进行求解, 然后根据前序优先级解的结果来更新后序服务的时间窗, 最终完成对整个地面服务问题的求解.

按照真实的地面保障服务流程, 选取一组不同优先级的服务, 具体如图 2 所示.

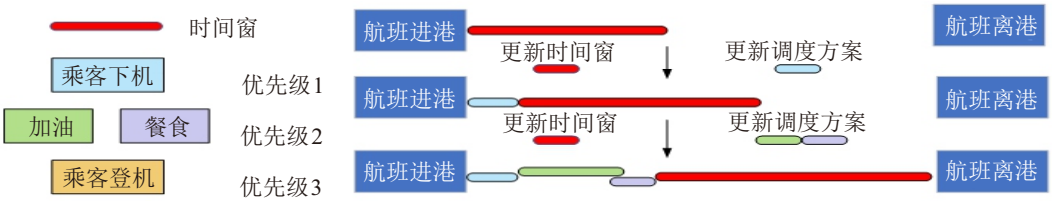

图2　子问题优先级分配示意图

根据不同服务的优先级, 将其分为 3 组子问题, 相同优先级的服务被视为同一组子问题, 具有相同的服务时间窗口, 不同航班的相同服务类型由一种车队统一进行服务. 在一系列的服务中, 根据航班过站的总时长与后序服务的服务时间和, 可以得出优先级最高服务的时间窗, 根据该类型服务结束的时间更新次序优先级的服务时间窗, 以此往复, 最终完成地面服务流程. 在该过程中, 地面服务流程被分解为相互独立的子问题, 由本文所提算法构建这些独立子问题的解.

1.2　模型假设

为了简化模型, 针对机场的环境做出如下假设:

1) 为保证航班准点率, 假设车队的车辆足够;

2) 假设机场特种车辆的行驶路线是固定的, 并且路径上无阻碍.

1.3　特种车群调度问题建模

首先, 采用无向图$G=(N,L)$来模拟实际机坪环境. 其中: $N=\{0,1,\dots,n,n'\}$为节点集, 对应机场的停机位和车场, 0 和n'均为车场, 用于区分车辆的起点和终点, 以避免车场的时间冲突; $L=\{(i,j)|i,j\in N\}$为边集, 实际表示为各个不同停机位之间的距离, 用以计算总路程以及车辆在不同停机位之间的转移时间. 另外, 为了区分起始点和航班, 定义$N'=\{1,2,\dots,n\}$表示待服务的航班序列. 定义操作集$M=\{1,2,\dots,n\}$, $\eta\in M$表示不同的服务. d_i^η表示位于节点i位置的航班对于服务η的需求量, 特殊地, d_0^η为停车场的需求量, 因此$d_0^\eta=0$. 对于$\forall\eta\in M$, 都有一个边(i,j)与其对应, 相应的距离可表示为c_{ij}^η, 将$c_{0n'}^\eta$点设置为无穷大, 避免起点与终点之间无意义的转移. 不同优先级的服务$\eta_1\succ\eta_2$可表示为服务η_1的优先级大于η_2, $(\eta_1,\eta_2)\in M$. 同一服务对应车队$V_\eta=\{1,2,\dots,n\}$, 设置同一车队中的车辆容量相同, 可表示为Q^η. 对于时间变量, t_i^η表示航班在位置i所接受η服务需要的时长, t_{ij}^η表示服务η对应的车辆从位置i到位置j的转移时长, T_{iv}^η表示用于服务η的车辆$v(v\in V_\eta)$服务航班i的开始时刻, c_{ijv}^η表示每辆车的服务距离. 此外, 定义$[A_i^\eta,B_i^\eta]$为航班i接受服务的时间窗, A_i^η为航班i接受η服务最早开始的时刻, B_i^η为航班i接受η服务最晚开始的时刻, 同一优先级的服务具有同样的时间窗. 为更好地描述不同优先级的服务时间窗, 对于航班i具有$U=\{1,2,\dots,u\}$优先级服务的时间窗定义为$[A_i^u,B_i^u]$. 决策变量$x_{ijv}^\eta\in\{0,1\}$确定车辆v服务航班i后是否服务航班j.

由于机场路径属于专有路径, 以总行驶里程作为规划目标, 在机场环境中同样反映行驶时间, 行驶里程的长短可以直接代表时间成本. 基于此, 目标函数选择车队完成服务所需的总路程长度, 最大程度上降低运营成本与节约旅行时间, 加快航班过站必须的服务流程. 目标函数公式化如下:

$$\min\sum_{\eta\in M}\sum_{v\in V_\eta}\sum_{(i,j)\in L}c_{ijv}^\eta x_{ijv}^\eta. \tag{1}$$

$$\text{s.t.}\sum_{j\in N}\sum_{v\in V_\eta}x_{ijv}^\eta=1,\ \forall i\in N',\ \eta\in M; \tag{2}$$

$$\sum_{j\in N'}x_{ijv}^\eta-\sum_{j\in N'}x_{jiv}^\eta=0,\ \forall i\in N'; \tag{3}$$

$$\sum_{i\in N'}\sum_{v\in V_\eta}x_{0iv}^\eta=\sum_{j\in N'}\sum_{v\in V_\eta}x_{jn'v}^\eta,\ \forall\eta\in M; \tag{4}$$

$$\sum_{i\in N\setminus\{0\}}\sum_{v\in V_\eta}x_{i0v}^\eta=\sum_{j\in N\setminus\{n'\}}\sum_{v\in V_\eta}x_{n'jv}^\eta=0,\ \forall\eta\in M; \tag{5}$$

$$\sum_{i\in N'}d_i^\eta\sum_{j\in N}x_{ijv}^\eta<Q^\eta,\ \forall v\in V_\eta,\ \eta\in M; \tag{6}$$

$$T_{iv}^\eta+t_i^\eta+t_{jv}^\eta\leqslant T_{jv}^\eta,\ \forall i,j\in N',v\in V_\eta,\eta\in M,x_{ijv}^\eta=1; \tag{7}$$

$$A_i^\eta\leqslant\ T_{iv}^\eta\leqslant B_i^\eta,\ \forall i\in N',\ v\in V_\eta,\ \eta\in M,\ x_{ijv}^\eta=1; \tag{8}$$

$$T_{iv}^{\eta_1}+t_i^{\eta_1}+\Delta_{\eta_1}\leqslant T_{iv}^{\eta_2},\ \forall i\in N',v\in V_\eta,(\eta_1\succ\eta_2)\in M. \tag{9}$$

式 (2) 确定航班的每一个服务都只由一辆车来完成. 式 (3) 控制去往航班i的车辆和离开车辆相同. 式 (4) 和 (5) 确保车辆在离开停车场为航班服务后回到停车场, 以及车辆的所有路径都从停车场开始至停车场结束. 式 (6) 保证了车辆的容量大于其服务

航班需求量的总和. 式 (7) 保证当车辆连续服务航班i与航班j时, 前序航班与后序航班的时间逻辑. 其中: T_{iv}^{η}表示航班i开始服务的时刻, t_i^{η}表示服务时长, t_{jv}^{η}表示车辆从航班i转移到航班j所需要的时长. 式 (8) 确保用于服务y的车辆v服务航班i的开始时刻位于该航班的时间窗内. 式 (9) 保证对于同一航班i, 不同优先级的服务η_1与η_2的开始时间顺序, $\eta_1 \succ \eta_2$表示服务η_1的优先级大于服务η_2, Δ_{η_1}表示开始服务时间的偏移量, 以此保证η_2服务的开始时间大于等于η_1服务的完成时间.

2 基于 Transformer 的 DRL 算法

2.1 MDP 模型设计

MDP 是一种离散状态的随机过程[10], 适用于序贯决策, 具有马尔可夫特性, 即未来状态仅依赖当前状态, 与历史无关. 它为强化学习提供了基础理论模型, 可用于抽象表述所有强化学习模型.

状态空间S: 出于状态特征全面性的考虑, 状态空间的选择舍弃了依赖专家经验而人为规定的具有实际意义的物理量, 转换为由 Transformer 框架对航班信息进行自动的提取. 具体为在t时刻, 由航班信息列表嵌入的静态特征与动态特征的叠加进行表示, 充分考虑了航班信息中蕴含的时间信息、相应需求以及前序航班的动态信息, 并综合当前服务车辆的类型、剩余容量以及最晚完成时间组成状态空间.

动作空间A: 动作集合为下一时刻选择服务的航班, 且不同类型的服务对应的服务车辆选择航班的策略相同.

状态转移概率P: 机场地面车辆调度问题中, 下一时刻的状态由当前时刻车辆状态以及航班信息所决定. 因此, 状态转移概率在所有时刻都为 1.

奖励R: 在时间步T完成所有的服务后, 能够得到航班的完整解决方案, 可表示为行驶轨迹$(S_1^{\eta}, S_2^{\eta}, \dots, S_t^{\eta})$. 通过对该解决方案中路径的求和可得到车队行驶的总里程数. 由于目标函数为总里程数最小, 奖励设定为总里程数的负值, 可表示为$R = -\sum_{t=1}^{T} S_t^{\eta}$.

2.2 基于 Transformer 架构的解集构建策略

文献 [11] 在解决车辆路径问题 (VRP) 中引入了注意力机制模型, 并且发现其模型性能大大优于传统的启发式算法. 此外, 传统的深度强化学习会将以往的学习经验保存, 并在选取下一个动作时将以往经验也作为参考依据[12]. 但是航班的服务需求时间和需求量受到天气和机械性能等不确定因素的影响, 在航班量较大时, 以往经验可能无法有效指导当前情况[13]. 而注意力机制能够捕捉输入数据的全局依赖关系, 使模型在面对动态变化的环境时能够更灵活地调整和更新其内部表示, 并且能够自动学习并提取输入数据中的重要特征, 避免传统方法中手动选择特征可能带来的偏差和不足. 基于此, 本文选择利用 Transformer 架构中的编码器与解码器分别对固定的状态特征以及动作选择策略进行更新. 训练得到的选择动作策略适用于航班所有的服务选择, 其表达公式化如下:

$$\pi(a_{t+1}^{\eta}) = \prod_{t=1}^{T} \pi(a_t^{\eta}|s_t^{\eta}) p(s_{t+1}^{\eta}|s_{t+1}^{\eta}, a_t^{\eta}). \tag{10}$$

其中: $\pi(a_{i+1}^{\eta})$为对于t时刻的状态s_t^{η}, 选择动作的策略; a_t^{η}为在t时刻被选择的动作, 具体表现为一个节点; $p(s_{t+1}^{\eta}|s_t^{\eta}, a_t^{\eta})$为状态转移概率, 在本节所构建的 MDP 模型中, 下一时刻状态由此时刻状态唯一确定, 因此$p(s_{t+1}^{\eta}|s_t^{\eta}, a_t^{\eta}) = 1$. 式 (10) 基于已固定的路径顺序和当前节点选择策略推导出下一服务航班. 与注意力机制模型解决 VRP 问题相比, 机场地面特种车辆调度面临更复杂的约束, 不仅需要多类型车辆的协同, 还需遵循航班时刻表并应对动态航班信息的挑战. 因此, 本文改进传统 Transformer 框架, 主要包括编码器和解码器两部分. 改进后的策略网络结构示意图如图 3 所示.

1) 编码器 Encoder.

编码器部分通过解析航班时刻表中航班的起降时间、相应的需求以及航班的停机位信息来生成所有可能访问的节点, 用以帮助解码器构建车辆的服务路径. 与文本翻译、时间序列预测等任务不同, 车辆调度任务的解是路径的集合, 在地图中这些节点不存在先后的顺序, 因此 Encoder 输入部分不需要引入位置编码环节. 同时, 与传统的旅行商问题 (TSP) 和 VRP 相对随机的任务地点不同[14], 机场地面服务的地点为该机场各个停机位, 其位置相对固定, 与文本翻译等任务有着相似的固定特征. 因此可以通过词嵌入的方式来计算各个节点包括停机位与停车场的位置.

对于航班时间与需求这类对时间高度敏感的参数, 常见的词嵌入方法便不再适用. 因为这类数据受到时间变化的影响, 通过词嵌入得到的词向量无法充分地表达时间特征. 为此, 本文考虑使用 LSTM 网络层进行嵌入[15], 以将输入特征投影到相应的维度, 从而生成节点的嵌入表示, 有效地对航班的需求以及时间特征进行跟踪. 随后将节点嵌入与航班时间、

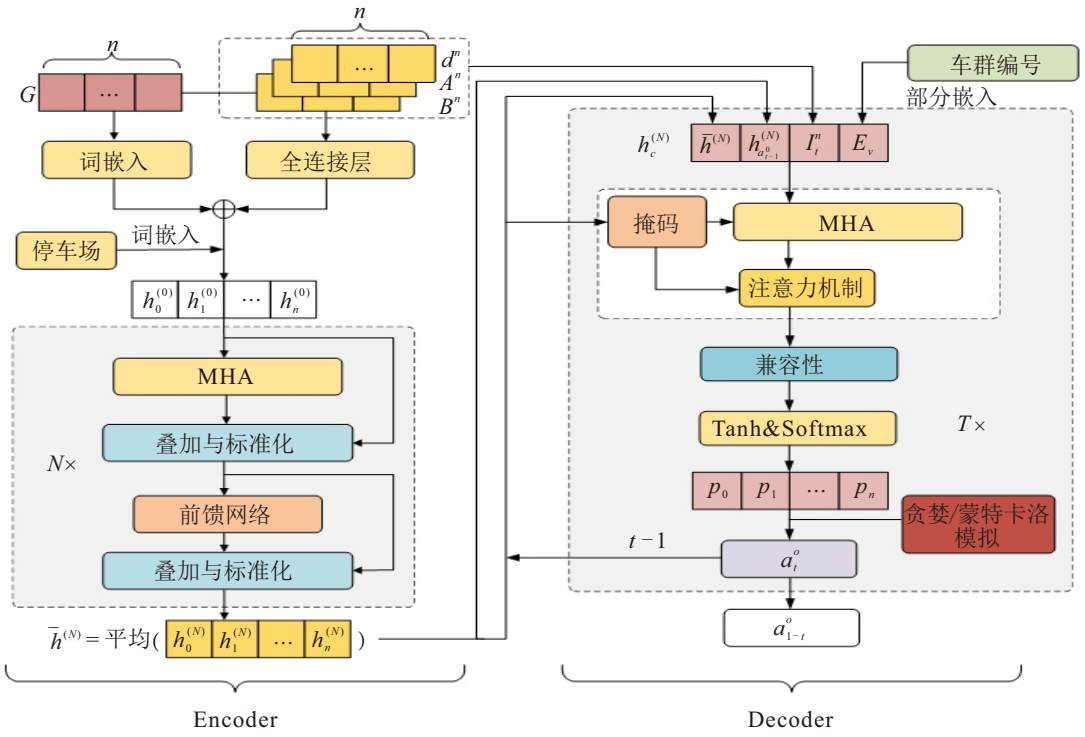

图3　策略网络结构图

需求投影信息相加得到初始嵌入, 表示如下:

$$h_i^{(0)} = \begin{cases} E_0, \ i = 0; \\ E_{gi} + \mathrm{LSTM}[(d^i, A_i, B_i), h_{t-1}, c_{t-1}], \\ \qquad i = 1, 2, \ldots, n. \end{cases} \tag{11}$$

其中: $h_i^{(0)}$为航班所在无向图G中节点i的嵌入表示, E_{gi}表示无向图G中待服务航班i的位置信息, (d^i, A_i, B_i)分别是节点i对某服务的需求量、最早开始时间以及最晚开始时刻, h_{t-1}是时间步$t-1$的隐藏状态, c_{t-1}是时间步$t-1$的细胞状态. 随后, 遵循传统的 Transformer 架构[16], 选择$N=3$次注意力层, 每个注意力层分为两个子层, 分别为多头注意力 (MHA) 层和前馈网络 (FF) 层, 每一层结束后分别利用跳跃链接与批归一化 (BN) 进一步处理各自的输出结果. 对于节点i, 以第l次更新为例, $l \in \{0, 1, \ldots, N\}$, 架构公式化如下:

$$\begin{aligned} \hat{h}_i^{(l)} = {} & \mathrm{BN}^{(l)}(h_i^{(l-1)} + \\ & \mathrm{MHA}^{(l)}(h_i^{(l-1)}; h_0^{(l-1)}, h_1^{(l-1)}, \ldots, h_n^{(l-1)})), \end{aligned} \tag{12}$$

$$h_i^{(l)} = \mathrm{BN}^{(l)}(\hat{h}_i^{(l)} + \mathrm{FF}^{(l)}(\hat{h}_i^{(l)})). \tag{13}$$

每一次注意力层更新, 其两个子层的参数一直保持同步更新. MHA 层和 FF 层的具体计算公式如下:

$$y_{jk}^h = \frac{(W_Q^h h_j)^{\mathrm{T}}(W_K^h h_k)}{\sqrt{d_h / H}}, \tag{14}$$

$$\hat{h}_j = \mathrm{MHA}(h_j; h_0, \ldots, h_n), \tag{15}$$

$$\begin{aligned} & \mathrm{MHA}(h_j; h_0, \ldots, h_n) = \\ & \sum_{h=1}^{H} W_O^h \Big[\sum_{k=0}^{n} \frac{e^{y_{jk}}}{\sum\limits_{k'} e^{y_{jk'}}} W_V^h h_k \Big], \end{aligned} \tag{16}$$

$$\mathrm{FF}(\hat{h}_j) = W_1^F \cdot \mathrm{ReLu}(W_0^F \hat{h}_j + b_0^F) + b_1^F. \tag{17}$$

其中: W_Q^h、W_K^h、W_V^h表示其中一个头的查询、键和值的线性变换矩阵; W_O^h是输出的线性变换矩阵, H表示 MHA 的头数量; W_0^F、W_1^F和b_0^F、b_1^F表示 FF 层中的线性投影值和偏执的可训练参数. MHA 层与 FF 层的具体架构可参考图 3. 初始嵌入经过N次更新后得到最终的节点嵌入$h_n^{(N)}$, 由各个节点的最终嵌入计算得到最终的图嵌入, 公式如下:

$$\bar{h}^{(N)} = \frac{1}{n+1} \sum_{i=0}^{n} h_i^{(N)}. \tag{18}$$

最终, 节点嵌入与图嵌入都作为航班以及地图信息输入到解码器来决策下一个待服务的航班.

2) 解码器 Decoder.

针对不同的场景, 文献 [17] 和文献 [18] 都将环境信息作为解码器的一部分加入到网络中, 作为求解的依据. 针对机场环境下的特种车调度问题, 本文将解码器的输入整体上分为航班信息输入与车辆信息输入两部分. 航班信息与车辆信息输入中, 编码器的整体图嵌入以及车辆的类型和相应的容量不受时间以及上一时刻服务节点的影响. 特别地, 在车辆特征嵌入时需要注意不同类型车辆所受到的约束条件不同, 若将所有车辆嵌入信息输入解码器将造成对

不同种类车辆的干扰而降低决策质量. 因此, 针对上述问题需要根据车辆服务类型引入部分嵌入的方式, 使得当前策略网络训练的服务类型与车辆嵌入类型一致. 如图 3 中的车辆特征嵌入部分所示, 公式化为

$$E_v = \text{Embedding}\left(\sum_{u=1}^{u} V^t_{\eta(\text{available})}\right). \tag{19}$$

其中: V^t_η表示t时刻对应的车辆服务类型; $V_{\eta(\text{available})}$表示$t$时刻所在时间窗内所有可服务的车辆嵌入, 避免了不同类型车辆造成的干扰.

当前服务车辆的信息以及已接受服务的航班路径嵌入由策略网络最终做出的决策序列决定. 同时, 解码器需要遵循时间顺序, 以上一时间步的服务节点为基础, 并结合后续所有可服务航班的相应服务需求及其时间窗口进行决策. 解码器以每一架航班服务调度完成作为一个时间步, 从$t=1$开始构建路径直到所有航班服务完毕. 除固定的图嵌入与车辆嵌入之外, 还需要上一时刻的服务节点嵌入$h^{(N)}_{a^\eta_{t-1}}$以及由服务车辆的剩余容量Q_t、最近该车辆服务航班的完成时间CT_t构成的当前服务车辆信息$I^n_t = \{Q, \text{CT}_t\}$, 更新规则如下:

$$Q_{t+1} = \begin{cases} 1, \ j_t = 0; \\ Q_t - d^\eta_{i_t}, \ j_t \neq 0; \end{cases} \tag{20}$$

$$\text{CT}_{t+1} = \begin{cases} 0, \ i_{t-1} = 0; \\ \max(\text{CT}_t + t^\eta_{i_{t-1}j_t}, a^0_j) + s^\eta_j, \ i_{t-1} \neq 0. \end{cases} \tag{21}$$

其中: j_t为t时刻服务的航班j, t_{t-1}为$t-1$时刻服务的航班i, s^η_j为航班j接受服务η的服务时间.

上述信息共同组成上下文节点以表示解码上下文信息, 表示为$h^{(N)}_c$, 长度设置为 128. 解码器中的掩码矩阵主要考虑时间上的服务顺序. 针对车辆调度任务, 已经被服务过的节点不能被车辆再次访问. 因此, 掩码矩阵中参数的更新规则如下:

$$m^t_j = \begin{cases} 0, \ j \notin a^\eta_{1:t-1}, d^0_j \leqslant Q_t, \\ \quad \max(\text{CT}_t + t^\eta_{i_{t-1}}, a^\eta_j) + s^\eta_j \leqslant b^\eta_j; \\ 1, \ \text{otherwise}; \end{cases} \tag{22}$$

$$m^t_0 = \begin{cases} 1, \ j_{l-1} = 0, \exists j \in N - \{0\}, m^t_j = 0; \\ 0, \ \text{otherwise}. \end{cases} \tag{23}$$

其中m^t_j与m^t_0分别表示t时刻掩码矩阵中所有待服务节点和车场的掩码参数, “1” 表示掩码矩阵中节点j已经在过往的服务中被选择, “0” 则相反. 掩码矩阵除遮掩已服务节点外, 还限制车辆剩余容量以及车辆最晚的完成时间. 遮掩规则基本遵循航班地面服务流程数学建模中的约束条件: 当车辆剩余容量大于节点需求时, 该节点不被遮掩, 反之则被遮掩; 当车辆服务节点j的前一节点i的最大完成时间以及转移时间小于节点j时间窗的最晚开始时间时, 该节点不被遮掩, 反之则被遮掩. 对仓库的遮掩规则与航班节点遮掩规则相似. 解码器 Decoder 首先经过上下文信息$h^{(N)}_c$的合成与掩码矩阵的图嵌入; 随后通过一次 MHA 处理, 再利用单头注意力机制计算每个节点相对于整体的注意力分数, 同时结合掩码后的上下文信息$h^{(N)}_c$进行调整; 最终经过兼容性模块C与 tanh函数后得到每一个节点的注意力分数$u_{(c)j}$, 通过 Softmax 计算后得到每一个节点被选择的概率分布$p_j \in \{p_0, p_1, \ldots, p_n\}$. 具体用公式表示如下:

$$u_{(c)j} = \begin{cases} C \cdot \tanh\left(\dfrac{W^Q h^{(N+1)}_c W^K h^{(N)}_j}{\sqrt{D_n/H}}\right), \ m^t_j = 0; \\ -\infty, \ m^t_j = 1; \end{cases} \tag{24}$$

$$p_j = \frac{e^{u_{(c)}(a^\eta_i)}}{\sum\limits_{j=0}^{n} e^{u_{(c)j}}}. \tag{25}$$

其中: W^Q和W^K分别表示上下文嵌入与单节点嵌入在单头注意力中的可训练的 Query 与 Key 矩阵, 其初始值分别为经掩码矩阵处理过的$h^{(N+1)}_c$和$h^{(N)}_n$; C为兼容性常数; $-\infty$在本研究中选择10^{-6}表示.

解码器选择下一个待服务节点的具体过程如下: 假设当前存在 4 个待服务节点, 解码器的输入由图嵌入、航班信息和车辆信息共同构成. 首先, 利用多头注意力机制整合上下文信息, 以获取整体的注意力分数; 随后, 通过单头注意力机制计算每个节点相对于整体的注意力分数, 并基于此选择最优节点, 逐步构建解集. 详细过程如下, 其中t表示时间步: 当$t=1$时, 经过解码器解码发现节点 3 对应的注意力分数最高, 因此选择节点 3; 当$t=2$时, 根据节点 3 的信息对解码器的输入进行更新. 除此之外, 还需要通过掩码矩阵对已经服务过的节点 3 的信息进行遮掩, 最终从剩余 3 个节点中依据注意力分数选取节点 1. 依此类推, 当$t=4$时, 解码器得到节点 4, 将每一时间步选择的节点进行组合, 得到该类型车辆的服务轨迹{3, 1, 2, 4}. 节点更多的实例解集过程与上述过程相同.

2.3 Transformer-DRL 算法训练过程

本文基础算法的训练过程是基于深度强化学习模型, 在最终神经网络框架输出结果后, Transformer 架构输出经过了 Softmax 算法处理, 转化为不同节点的概率分布. 基于此, 设计两种选择下一动作的方

式: 利用贪婪策略 (Greedy), 简单地选择概率最大的节点作为下一服务对象; 利用蒙特卡洛模拟 (MCS), 根据给出的概率分布进行一定次数的随机采样, 并计算每一次采样结果的平均值, 选择结果最优的节点作为下一服务对象. 将图策略网络架构定义为基线网络. MCS 以计算量为代价对可能的结果进行分析, 增强了最终解集的可靠性.

基线网络与策略网络借鉴了 DQN 模型中目标网络与在线网络的思想[8], DQN 会初始化一个Q网络和一个目标Q网络, 每隔一定的训练时间将Q网络的参数复制给目标Q网络, 通过这种方法来避免Q值更新过快导致的目标值波动, 以此来减缓训练的不稳定性. 此外, 网络的更新采用t-test 的方法[19]. 具体的设置思路如下: 在初始时两个网络参数相等, 基线网络参数不断地更新, 当最新的基线网络参数显著优于策略网络参数时, 将基线网络参数更新到策略网络中, 差异程度取决于参数$\alpha = 5\,\%$. 通过两个网络的参数不断迭代, 将有效解决神经网络过拟合的问题, 以及避免不同种类服务嵌入相互干扰, 找到更高质量的调度方案. 由于基线网络在计算奖励时实时地反映了当前航班接受服务的种类, 与策略网络仅反映整体优化结果有一定的差异, 且对航班数据进行特征提取时的网络参数实时更新, 导致基线网络与策略网络在验证集上得到的奖励值存在差异, 算法利用这一差异对网络参数进行更新. 算法中的 loss 函数具体公式如下:

$$\frac{1}{N_b}\sum_{m=1}^{N_b}(R_m - R_m^L)\nabla_\theta \log \pi_\theta(s_{t,m}|s_{1,m}). \tag{26}$$

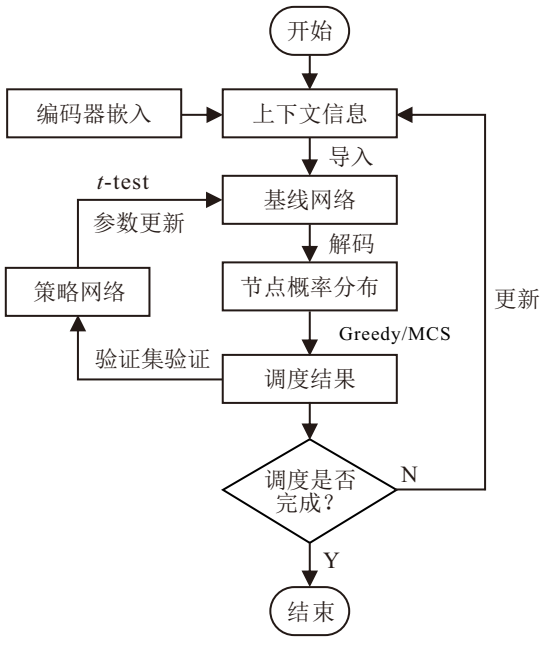

图4　算法流程

式 (26) 中考虑了同一个航班所有的服务过程. 为了淡化不同服务之间的影响, 将 Batch size 中对于基线网络与策略网络的奖励偏差进行了平均. 此外, 对网络参数的梯度做了 log 运算, 使其变化得更加平滑, 有助于模型网络的训练. 图 4 为 Transformer-DRL 算法训练流程, 直观地展现了模型的训练过程.

3　实验验证

本文使用基于 Transformer-DRL 的算法求解机坪特种车群协同调度问题, 实验环境为 python3.9, 框架为 torch1.12.1, 处理器为 AMD Ryzen 75800H, 内存为 24.0 G RAM.

3.1　实验数据准备

本文以天津滨海国际机场为例, 主要涉及 64 个停机位. 针对航班数据, 对真实航班起降时间进行采样, 生成 20、50、90 三种数量的航班实例, 分别称为 F20、F50、F90. 由于本文所针对的问题是多种类特种车辆对航班的协同保障, 需要综合考虑航班对不同种类特种车辆的需求. 为简化模型训练, 只考虑一种飞机类型, 并参考文献 [20] 得到本文涉及地面保障服务的特种车辆数据.

摆渡车与客梯车在解码器求解过程中可视为同一组车辆类型, 其中客梯车没有服务容量限制, 路径上可服务多架航班. 经过调研, 我国现有大中型机场大多在机位处铺设了输油管线, 机场均将传统的罐式加油车换装为管线式加油车, 摆脱了容量限制, 因此加油车容量可默认为一个极大的数字. 除此之外, 对不同航班对应的需求以及车辆参数进行调整: 将航班需求与车辆容量嵌入神经网络时, 为统一量纲便于计算, 需要进行归一化计算, 车辆容量初始为 1, 航班不同服务的需求量从$[0.1, 0.2, 0.3]$中随机选择; 考虑到满足航班时间窗需要车辆数量足够大, 每一种服务对应的特种车辆数量分别为 20、40、60 以保证航班的准点率.

3.2　实验结果分析

1) 算法训练过程可视化.

本文分别模拟了算例规模为 20、50、90 的实例, 对算法的训练过程进行可视化, 训练曲线如图 5 所

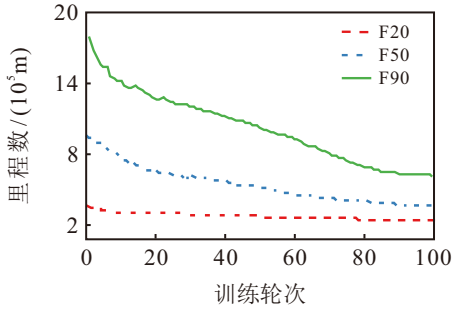

图5　训练曲线

示. 随着迭代的进行, 策略网络对基线网络的参数不断更新, 更容易找到一个更优的基线模型参数加速深度强化学习算法的收敛. 当迭代次数接近 100 后, 训练结果变得平缓, 表明其收敛到一个稳定的结果.

2) 消融实验.

通过本实验, 验证本文所用 Transformer 结构中的几个关键设计点来讨论其性能. 实验分为两部分: 一部分验证不同的损失函数, 另一部分探讨不同时间优先级嵌入方式, 并将这些结构应用于 F100 实例中进行验证.

损失函数: 本文在每个子问题上使用的调度策略在各个车队之间是共享的, 而对应的损失函数计算的是每个车队的独立成本. 由于机场特种车辆调度问题需要多种特种车辆之间的协调, 式 (21) 所示的损失函数在策略学习过程中可能会忽视环境的全局特征或不同车队之间的相互影响. 基于此考虑, 对本文提出的损失函数进行扩展, 具体定义如下:

$$L_1 = \frac{1}{N_b}(R - R^L)\prod_{m\in M}\pi_\theta(s_{t,m}|s_{1,m}). \tag{27}$$

式 (27) 中L_1损失函数以全体车队的总行驶里程来优化学习策略. 其中: R表示本文模型计算后的车队总体调度里程, R^L表示基线模型的全部车队调度总里程. 此外扩展的第 2 个损失函数的定义如下:

$$L_2 = \frac{1}{N_b}\sum_{m\in M}(aC_{\text{sig}} + (1-a)C_{\text{total}})\nabla_\theta \log \pi_\theta, \tag{28}$$

$$C_{\text{sig}} = R_m - R_m^L, \tag{29}$$

$$C_{\text{total}} = R - R^L. \tag{30}$$

L_2损失函数同时考虑总体车队的行驶里程和单个车队的行驶里程, 将两者以不同的权重求和, 充分考虑局部与全局特征. 其中$a = 0.95$.

时间特征嵌入方式: 对于机场环境中受时间影响的特征, 本文使用 LSTM 网络作为时间信息的嵌入方式. 为了验证 LSTM 网络的作用, 本文采用不同的嵌入方式进行处理. 具体分为 3 种方式, 包括去除时间特性即不嵌入时间窗数据, 利用传统的线性投影进行嵌入, 以及使用门控循环单元 (GRU) 进行嵌入. 不同损失函数与不同嵌入策略的结果如图 6.

从图 6 可以观察到: 以L_1作为损失函数时, 模型在给定的迭代次数内未能收敛; 相较之下, 结合了每个车队里程和总体里程的L_2损失函数表现更好, 但其最终收敛效果仍不及本文所采用的损失函数. 两种损失函数的计算结果表明, 综合考虑两种里程数的损失函数在计算中可能引入了额外的噪声, 干扰了网络的决策过程, 从而导致收敛效果变差.

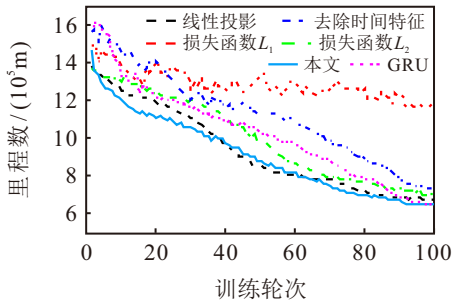

图6 消融实验训练曲线

对于时间嵌入方式, 去除时间窗信息嵌入后的迭代曲线显著劣于包含时间窗特征的模型. 此外, 针对不同的时间嵌入网络, 包括线性投影方式和 GRU, 从图 6 可以看到, 线性投影方式对于网络最终的收敛并没有提升, 而 GRU 网络与线性投影方式结果相近, 但收敛速度较本文使用的 LSTM 网络要逊色. 表 1 记录了在测试集上不同损失函数和嵌入方式的结果.

表1 测试集消融实验结果

	本文算法(MCS)		本文算法(Greedy)	
	里程数 / m	时间 / s	里程数 / m	时间 / s
L_1损失函数	1 134 792	12	1 198 248	14
L_2损失函数	709 784	14	1 071 552	13
去除时间特征	766 872	25	977 440	26
线性投影	671 320	15	727 400	13
GRU	651 112	15	1 007 328	15
本文算法	647 048	13	664 345	13

此外, 为了探讨掩码规则在提高采样效率和改善结果质量方面的作用, 本文以 F50 算例为基础, 统一采用 Greedy 方法进行实验. 实验设计中, 对是否采用掩码规则进行了系统的对比分析, 结果见图 7.

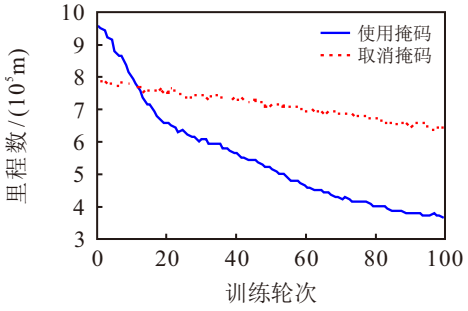

图7 掩码消融实验曲线

从图 7 中曲线的趋势可以看出, 本文提出的掩码规则相比于仅依赖 Transformer 架构的输出并采用 Greedy 方法进行决策, 具有更快的收敛速度. 同时, 使用该掩码规则的结果在收敛效果上也显著优于单纯的 Greedy 方法. 这表明, 所提的掩码规则不仅能够有效加速模型的训练过程, 还能够提升其最终的表现质量.

3) 对比实验.

将本文算法与常见的解决 VRP 问题的方法在不同规模的算例上进行比较. 选择目标函数值、运行时间两个参数作为评价指标. 为衡量不同算法在测试集上的稳定性, 引入一种新的评价指标 —— 差异值, 用于衡量算法在测试集中的平均结果与最佳结果的差距, 差距越大代表算法在面对测试集的效果越不稳定. 比较结果如表 2 所示.

表2　不同算法的对比数据

方法	F20			F50			F90		
	目标值 / m	差异 / %	时间 / s	目标值 / m	差异 / %	时间 / s	目标值 / m	差异 / %	时间 / s
CPLEX	208 591	0	0.01	746759	99.25	0.01	1955333	242.94	0.01
模拟退火	293097	51.23	0.01	677745	61.13	0.01	1217162	92.35	0.01
遗传算法	396341	85.97	0.01	815037	107.10	0.01	1522541	126.54	0.01
最邻近节点插入	367592	79.64	2.86	767694	138.62	27.35	1401742	161.49	0.0033
随机节点插入	405984	92.58	2.12	800393	142.87	0.022 42	1548892	171.27	0.00255
节约算法	289247	21.37	0.09	538457	34.50	0.24	792914	33.98	0.48
DQN	334290	18.84	6	615248	0.25	11	812740	12.11	18
MAPPO	315059	1.34	10	552496	9.07	13	777327	5.37	19
本文算法(MCS)	223735	0.42	3.64	364624	0.11	4.52	599632	0	6.38
本文算法(Greedy)	230432	2.59	3.52	375916	5.22	4.11	601453	6.98	4.97

CPLEX 方法采用了精确求解的分支定界算法, 在面对小规模算例时, 在 10 min 内能够取得较为精确的最优解. 在算例规模不断扩大时, 由于问题的 NP 难特性导致解集空间维度以指数形式增长, 基于精确求解方法的 CPLEX 算法在短时间内无法找到较为满意的解[21]. 元启发式方法, 通过构造适应度函数并结合专家经验进行算子设计, 有规律地对解集空间进行探索, 在较短的时间内得到了较为理想的结果, 且随着适应度函数以及算子设计的精细化, 优化目标也会不断提升[22]. 对于经典的插入算法与节约算法, 由于其较为简洁的解集构造过程, 有着相对较低的算法复杂度, 能够在极短的时间内得到结果.

本文所用为端到端架构, 利用深度强化学习对模型参数进行更新. 在处理小规模算例时, 由于节点信息较少, 对环境探索的不够充分, 导致效果略差于利用精确算法求解的 CPLEX. 随着算例的不断增加, 深度强化学习算法能够对状态空间进行充分的探索与学习, 并利用经验得到比传统的算法都要优异的结果, 且经过预训练直接调用训练好的神经网络, 用时较短. 由于本文端到端架构中对车群嵌入采取了部分嵌入的方式以及在模型训练时引入了基线网络与策略网络并行的训练方式, 模型的求解时间略大于其他算法. 然而, 在解集质量方面, 随着算例的增加, 优势逐渐扩大.

使用强化学习框架下的 DQN 和 MAPPO(multi-agent proximal policy optimization) 算法来求解调度策略时, 虽然同为端到端架构, 但是缺乏本文算法中针对环境的掩码处理和网络的更新策略, 在面对不同算例时无论是计算时间还是稳定性都会劣于本文所提出的算法.

针对本文所提出的两种节点选择策略, 通过比较 MCS 与 Greedy 策略发现, MCS 策略通过若干次采样不断地在验证集上比较所有节点的结果, 消耗了比 Greedy 策略更多的时间, 但是相对于 Greedy 策略在最终目标上有一定的提升.

表 3 展示了 F50 算例下各车队的调度策略. 表 3 中的数据表示对应编号车队的航班服务顺序, 以及完成服务的航班数量.

表3　F50 算例下车队的调度方案

车队编号	调度方案	航班服务完成数量
1	0→28→0→10→9→25→23→2→30→0→19→12→44→7→26→ … →43→20→0→35→0	50
2	0→23→26→38→31→0→48→20→14→33→1→0→42→29→30→ … →40→11→0→0→0	50
3	0→38→27→12→32→16→31→4→0→43→1→2→49→34→0→8→ … →0→5→0→22→0	50
4	0→36→8→11→14→34→40→0→28→15→30→26→0→18→7→0→5→ … →12→0→0→0	50
5	0→20→24→19→33→0→27→47→9→34→8→0→42→28→11→48→ … →0→36→31→0	50
6	0→49→31→21→32→10→44→0→7→36→0→41→9→38→0→33→ … →40→ 0→19→0	50
7	0→41→13→19→0→30→29→8→23→4→0→36→49→0→43→31→35→ … →0→7→0→42	50
8	0→22→44→35→29→0→3→1→50→7→16→0→2→43→10→5→31→20→ … →41→0→0→0	50
9	0→23→37→46→20→27→9→18→47→29→25→0→40→38→33→14→ … →13→0→0→0	50
10	0→50→32→17→9→0→26→36→30→23→1→0→45→24→8→3→19→ … →16→0→21→0	50

4) 泛化研究.

通过对 F20、F50、F90 的实例进行研究发现, 已经训练好的策略网络分别在其生成的验证集上取得了良好的结果. 为进一步验证经过训练的模型的可靠性, 又额外生成规模为 200 和 300 的实例对模型解决大规模问题的能力进行评估, 结果如表 4 所示.

表4 大规模实例验证结果

策略	F200			F300		
	目标 / m	差异 / %	时间 / s	目标 / m	差异 / %	时间 / s
MCS	1729232	0.003	9.67	2408950	0	16.52
Greedy	1766287	6.271	7.14	2678607	4.859	9.11

由表 4 可以发现, 在算例规模不断扩大时, MCS 策略选择的结果相比 Greedy 策略更加稳定, 且运算时间可以接受, 能够认为在一定程度上, MCS 策略比 Greedy 策略更加优秀.

5) 航班的实时性分析.

为验证本文方法在面对动态的航班时间时的处理能力, 利用变化的航班数据进行实验验证, 并采用机会约束方法[5], 对航班的到达时间添加 [0, 15] 的偏差值以模拟真实的机场工作场景. 由于机场车辆调度的马尔科夫特性, 当前时刻已被安排服务的航班不受到影响. 以 F50 为例, 实验结果如表 5 所示.

表5 实时航班信息测试

策略	静态航班时间		动态航班时间	
	目标值 / m	时间 / s	目标值 / m	时间 / s
MCS	368041	4.53	376541	4.99
Greedy	370683	4.11	380638	4.67

引入动态的航班时间后, 目标值方面, 航班时间变化造成后续的服务路径改变, 打破了原有规划的最佳路径, 因此总行驶长度有所增加. 求解时间方面, 航班时间变化造成编码器的输入发生变化, 图嵌入随着航班时间变化而变化, 算法求解时间有所增加, 但仍远远低于传统方法面对静态问题时的求解时间. 实验结果表明, 基于 Transformer-DRL 算法在面对动态的调度任务时有良好的表现.

4 结 论

本文提出了一种基于 Transformer 的深度强化学习算法解决地面车群协同调度问题. 首先, 通过不同的服务优先级顺序将复杂的多类型车辆调度问题分解为独立的车辆路径问题, 并用构建的策略网络框架对各个独立问题进行求解. 其次, 为避免不同服务之间的相互干扰以及不同种类车辆特征的融合, 提出了两个初始参数完全相同的两种策略网络进行更新, 从结果来看有效地避免了不同服务之间的影响. 提出了部分嵌入的方法, 根据不同优先级的服务对车辆特征进行有选择的嵌入, 解决了车辆特征融合的问题. 最终, 对真实数据进行采样并扩充, 与经典的启发式算法以及元启发式算法在不同规模的实例上进行了对比并详细分析其结果, 证实本文所提算法在面对复杂的车辆路径问题时优于启发式算法. 随后对模型的泛化能力以及应对动态航班的实时性进行了分析.

综合上述结果, 可以证实本文所提算法在解决机场地面特种车群协同调度问题上取得了成功.

参考文献 (References)

[1] Du Y Q, Zhang Q, Chen Q S. ACO-IH: An improved ant colony optimization algorithm for Airport Ground Service Scheduling[C]. 2008 IEEE International Conference on Industrial Technology. Chengdu, 2008: 1-6.

[2] Du J Y, Brunner J O, Kolisch R. Planning towing processes at airports more efficiently[J]. Transportation Research Part E: Logistics and Transportation Review, 2014, 70: 293-304.

[3] Li Q W, Bi J, Li Z Y. Research on ferry vehicle scheduling problem within airport operations[C]. 2017 10th International Symposium on Computational Intelligence and Design. Hangzhou, 2017: 248-251.

[4] Guo W A, Xu P, Zhao Z, et al. Scheduling for airport baggage transport vehicles based on diversity enhancement genetic algorithm[J]. Natural Computing, 2020, 19(4): 663-672.

[5] Zhu S R, Sun H J, Guo X. Cooperative scheduling optimization for ground-handling vehicles by considering flights' uncertainty[J]. Computers & Industrial Engineering, 2022, 169: 108092.

[6] 余明晖, 周鼎新, 汤皓泉. 基于 DQN 的机场地服人员动态排班研究[J]. 华中科技大学学报: 自然科学版, 2022, 50(11): 66-71.
(Yu M H, Zhou D X, Tang H Q. Study on airport ground staffs dynamic scheduling based on DQN[J]. Journal of Huazhong University of Science and Technology: Natural Science Edition, 2022, 50(11): 66-71.)

[7] Wu Y X, Zhou J N, Xia Y W, et al. Neural airport ground handling[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(12): 15652-15666.

[8] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[9] 姜伟华, 张文静, 袁琪, 等. 基于时间窗的机场地面保障车辆动态调度[J]. 科学技术与工程, 2024, 24(3): 1283-1291.
(Jiang W H, Zhang W J, Yuan Q, et al. Dynamic scheduling of airport ground support vehicles based on time window[J]. Science Technology and Engineering,

2024, 24(3): 1283-1291.)

[10] 李海峰, 杨宏安, 盛梓茂, 等. 基于 MAPPO 的多无人机协同分布式动态任务分配[J]. 控制与决策, 2025, 40(5): 1429-1437.
(Li H F, Yang H G, Sheng Z M, et al. Multi-uav collaborative distributed dynamic task assignment based on MAPPO [J]. Control and decision, 2025, 40(5): 1429-1437.)

[11] 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.

[12] Zhu K, Zhang T. Deep reinforcement learning based mobile robot navigation: A review[J]. Tsinghua Science and Technology, 2021, 26(5): 674-691.

[13] 张文义, 唐雨拉尔, 王旭兰, 等. 考虑双时间窗特性的机场多车型摆渡车调度优化[J]. 北京航空航天大学学报, DOI: 10.13700/j.bh.1001-5965.2023.0579.
(Zhang W Y, Tang Y L, Wang X L, et al. Scheduling optimization of airport multi-vehicle ferries considering dual time window characteristics [J]. Journal of Beijing University of Aeronautics and Astronsutics, DOI: 10.13700/j.bh.1001-5965.2023.0579.)

[14] Kou S H, Golden B, Poikonen S. Estimating optimal objective values for the TSP, VRP, and other combinatorial problems using randomization[J]. International Transactions in Operational Research, 2024, 31(5): 3443-3458.

[15] Maged M, Mostafa E, Hesham H. Improving flight delays prediction by developing attention-based bidirectional LSTM network[J]. Expert Systems with Applications, 2024, 238: 121747.

[16] Yue M, Ma S H. LSTM-based transformer for transfer passenger flow forecasting between transportation integrated hubs in urban agglomeration[J]. Applied Sciences, 2023, 13(1): 637.

[17] Bogyrbayeva A, Yoon T, Ko H, et al. A deep reinforcement learning approach for solving the traveling salesman problem with drone[J]. Transportation Research Part C: Emerging Technologies, 2023, 148: 103981.

[18] Tang M C, Zhuang W C, Li B B, et al. Energy-optimal routing for electric vehicles using deep reinforcement learning with transformer[J]. Applied Energy, 2023, 350: 121711.

[19] 陈维兴, 李业波. 基于 DQN 的机场加油车动态调度方法研究[J]. 西北工业大学学报, 2024, 42(4): 764-773.
(Chen W X, Li Y B. Research on dynamic scheduling method of airport refueling truck based on DQN[J]. Journal of Northwestern Polytechnical University, 2024, 42(4): 764-773.)

[20] Xu Y Q, Fang M, Chen L, et al. Reinforcement learning with multiple relational attention for solving vehicle routing problems[J]. IEEE Transactions on Cybernetics, 2022, 52(10): 11107-11120.

[21] Zhou J N, Wu Y X, Cao Z G, et al. Learning large neighborhood search for vehicle routing in airport ground handling[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(9): 9769-9782.

[22] 薛阳, 倪大斌, 卢秋红, 等. 基于 PGWO 算法的移动机器人路径规划[J]. 控制与决策, 2025, 40(4): 1395-1401.
(Xue Y, Ni D B, Lu Q H, et al. Mobile robot path planning based on PGWO algorithm[J]. Control and Decision, 2025, 40(4): 1395-1401.)

作者简介

陈维兴 (1981−), 男, 副教授, 硕士生导师, 主要研究方向为嵌入式和网络系统、AI 系统、群智感知系统、智慧机场, E-mail: cw007x130@vip.163.com;

李晨辉 (2001−), 男, 硕士生, 主要研究方向为多智能体调度, E-mail: lch1127133364@163.com;

李业波 (1998−), 男, 助理工程师, 硕士, 主要研究方向为空管自动化、机场地面保障车辆调度, E-mail: 936237348@qq.com.