

基于分层深度强化学习的O2O取送货动态调度

高明¹, 陈明浩¹, 唐加福¹, 邹广宇², 许欣¹

(1. 东北财经大学管理科学与工程学院, 辽宁 大连 116012;

2. 大连理工大学电气工程学院, 辽宁 大连 116024)

摘要: 针对O2O即时配送调度中需求波动、路况不确定及实时性挑战, 提出一种分层深度强化学习方法. 上层智能体不断学习动态变化的历史订单及路况信息, 进行骑手任务分配; 下层专注于各骑手并单后的路径优化. 通过全局奖励函数, 在分层智能体间纵向传递全局优化信号, 并在多个滚动调度区间内横向协调长期平均目标. 在仿真平台中对大连市某外卖平台的真实和模拟订单进行了多场景实时调度实验, 验证了方法在滚动调度中兼顾长期目标的优越性和分层求解的高效性, 为即时配送服务提供了兼具成本效益和服务质量的优化调度解决方案.

关键词: 分层深度强化学习; O2O即时配送; 动态取送货问题; 交通仿真

中图分类号: TP273 文献标识码: A 文章编号: 1000-5781(2026)01-0064-13

doi: 10.13383/j.cnki.jse.2026.01.005

Dynamic scheduling of O2O pick-up and delivery based on hierarchical deep reinforcement learning

Gao Ming¹, Chen Minghao¹, Tang Jiafu¹, Zou Guangyu², Xu Xin¹

(1. School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China;

2. School of Electrical Engineering, Dalian University of Technology, Dalian 116025, China)

Abstract: To solve O2O delivery challenges like changing demands and real-time requirements, this study proposes a two-layer deep reinforcement learning method. The upper-layer agent handles order allocation by learning from historical orders and road conditions. The lower-layer agent focuses on optimizing delivery routes for riders. A global reward system connects both layers to share optimization signals. This design helps balance long-term goals across multiple scheduling periods. In the simulation platform, real and simulated orders from a certain food delivery platform in Dalian are subjected to multi-scenario real-time dispatching experiments. This verifies the superiority of the method in considering long-term goals in rolling scheduling and the efficiency of hierarchical solving, providing an optimized dispatching solution that is both cost-effective and service-quality-oriented for immediate delivery services.

Key words: hierarchical deep reinforcement learning; O2O instant delivery; dynamic pickup and delivery problem; traffic simulation

收稿日期: 2023-01-04; 修订日期: 2025-04-07.

基金项目: 国家自然科学基金资助项目(72293563); 辽宁省自然科学基金资助项目(2024-MS-175); 辽宁省教育厅基本科研资助项目(JYTZD2023050); 大连市科技人才创新支持计划资助项目(2022RG17); 辽宁省教育厅研究生科研创新专项资助项目(DUFYJS24035); 辽宁省重点研发计划资助项目(2024JH2/102400020).

*通信作者

1 引言

O2O(online to offline)即时配送已成为现代生活的重要组成部分,随着市场规模的不断扩大,平台面临着合理调配资源,高效完成订单的运营挑战.通过大数据和优化技术进行调度,减少订单超时和配送成本等,成为产业界和学术界共同关注的热点问题^[1].

诸多学者将该问题看作带软时间窗的取送货调度问题(pickup and delivery problem with soft time windows, PDPSTW)^[2,3],PDPSTW是车辆路径问题(vehicle routing problem, VRP)的一个变体^[4],VRP是一个NP难问题,面临着解空间巨大,求解效率低的挑战,在多项式时间内难以提供有效最优解.因此,一些求解PDPTSW的优秀的精确算法^[5,6]和启发式算法^[7,8]不再适用于具有时间紧迫性的大规模即时配送调度问题.此外,即时配送调度还面临着订单动态随机出现、需求波动性强以及路况实时不确定等问题.构建能够适应需求时空波动性、路况实时不确定性并满足决策严格时效性的动态调度方法,是突破传统PDPSTW求解局限的关键研究问题.

针对解空间巨大的问题,Li等^[9]通过先对订单进行聚类,再对聚类后的小规模订单进行路径优化,既减小了问题规模,又减少了计算时间.然而,该方法通常仅能实现单次调度,需等待当前订单全部配送完成后才能对剩余订单重新调度,无法在配送过程中动态调整骑手的任务分配与路径规划,造成资源浪费.Liang等^[10]设计了一种混合遗传算法和插入算法,能够将新订单不断插入车辆递送路径进行滚动调度,从而实现资源的最优利用^[11,12].然而,将动态调度拆解成一系列静态问题的滚动调度难以兼顾长期目标.此外连接餐厅和消费者的骑手对配送质量具有重要影响,如何改善其工作环境对维持平台稳定性和提高配送服务质量具有重要意义^[13].Jiang等^[14]通过减小骑手的利润效率方差,旨在减少骑手之间的收入差异,从而提高骑手满意度.凌帅等^[15]在此基础上考虑了骑手安全,通过遗传算子的设计,实现多目标协同优化.但此类启发式方法主要针对小规模算例场景,在大规模场景下,容易陷入局部最优,且算子很大程度上依赖于人类的专业知识和经验,需要不断地监测和调整策略,否则会导致该方法的效果不佳.

在深度学习(deep learning, DL)和强化学习(reinforcement learning, RL)的快速发展的支持下,基于学习的方法已被广泛应用于优化调度问题中^[16].RL能够利用离线经验,在学习过程自适应地改善策略.而DL能够对环境状态进行感知,提高配送效率.Li等^[17]将二者进行结合,使用深度强化学习方法(deep reinforcement learning, DRL)首次对取送货问题进行了求解,并验证了它具有比传统启发式方法更好的性能.在PDPSTW方面,Zou等^[18]利用开源交通模拟软件SUMO(simulation of urban mobility)构建了整个O2O平台的模拟模型,使用双深度Q网络(double deep Q-network, DDQN)根据订单和骑手状态生成分配动作,通过奖励机制引导智能体学习,使订单完成时间最短,并验证该方法的有效性.陈彦如等^[19]利用DRL感知需求不确定性并优化任务分配,同时考虑骑手间的相互影响,使用启发式快速生成路径规划方案,该方法具有良好的泛化性和适应性.然而,这些方法未充分考虑现实场景中复杂路况对配送的干扰,这些因素会直接影响骑手的调度与路线规划.此外,下层简单使用启发式算法对路线进行调整,虽然能够快速生成路径,但在处理复杂多变的路况时,可能无法找到最优解,导致路径规划不够精细和高效.通过在路径优化中引入DRL,有望进行改善.

综上,为了应对大规模PDPSTW求解复杂度高,传统算法对不确定环境适应性差、易陷入局部最优,对真实配送环境耦合不足的问题,本文提出了一种分层DRL调度算法SAC-PNLNS(soft actor-critic-pointer network-large neighborhood search),在任务分配和路径优化均使用智能体进行决策,利用其动态感知能力,适应订单需求和路况信息的不断变化,持续优化策略,提高横向优化结果.综合考虑订单平均服务距离、平均超时惩罚和骑手公平性(骑手效益方差),为上下层智能体设计奖励函数,引导智能体各自优化改进策略.此外,设计了全局奖励函数,进行纵向智能体间信号传递,避免优化目标冲突导致的局部优化,实现平台、客户、骑手三个主体的总体综合收益的最优.最后在交通仿真平台中对真实订单进行调度模拟,验证了所提算

法的优越性和适应性.

2 问题介绍

在O2O即时配送平台上, 由于订单需求的显著波动和随机性, 以及客户对一小时内完成配送的严格要求, 平台面临着即时响应订单的挑战. 同时, 备餐时间的随机性和路况的不确定性也对配送效率产生显著影响. 在骑手资源有限且存在背包容量限制的条件下, 本文研究了如何在实时任务分配和路径规划中考虑这些不确定性因素, 以减小订单的平均服务距离, 降低超时成本, 提高平台和客户的满意度, 同时减小骑手效益方差, 提高骑手的满意度, 从而优化整个配送流程.

具体地, 本文使用有向图 $G = (N, B)$ 表示平台配送交通网络. N 表示骑手配送过程中可能经过的节点集合, 包含配送服务中心 N_{init} 、商家节点集合 J , 和客户节点集合 K . B 是 N 中所有节点两两间组成的边集合, 每条边 (n, n') 对应一个固定距离成本 $d(n, n')$, 以及路况干扰下的不确定骑手速度影响因子(路况影响因子) $\delta(n, n')$. 考虑同质骑手, 每个骑手拥有相同的背包容量限制 C , 额定速度 V . 初始时刻, 所有骑手均位于 N_{init} . 在运营周期 $[0, T]$ 内, 顾客订单 i 动态到达, 系统需要进行任务分配, 并优化骑手路线.

已到达订单 $i \in I$ 包含以下信息: 订单号 i , 商家节点 $j_i \in J$, 下单时间 t_i , 最早取餐时间 l_i , 最晚柔性送达时间 f_i (为下单后一小时对应时间), 以及下单客户 $k_i \in K$. 其中最早取餐时间需要考虑备餐时间 t_p^i , 假设 $t_p^i \sim N(\mu_1, \lambda_1)$.

对于完整骑手集合 H , 骑手 $h \in H$, 包含以下信息: 背包容量限制 C , 骑手当前携带订单量 c_h . 骑手当前所在位置 $n_h \in N$, 当前骑手总收益 e_h . 系统进一步结合骑手信息进行任务分配, 在考虑容量约束下, 将任务分配给可接单骑手. 在某个时刻骑手的路线计划可以表示为

$$\theta_h = \left(\begin{array}{l} (n_{\text{init}}^h, d(n_{\text{init}}^h)), (n_1^h, s(n_1^h), a(n_1^h), d(n_1^h)), \dots, \\ (n_\xi^h, s(n_\xi^h), a(n_\xi^h), d(n_\xi^h)), \dots, (n_o^h, s(n_o^h), a(n_o^h)) \end{array} \right), \quad (1)$$

其中 n_{init}^h 表示表示骑手 h 的出发地, $d(n_{\text{init}}^h)$ 表示 h 出发的时刻, 通常为其接到第一个订单的时刻. 对已访问过的节点 n_1^h , 路径中包含该节点对应的服务时间 $s(n_1^h)$, 骑手实际到达时间 $a(n_1^h)$, 和实际离开时间 $d(n_1^h)$. 对于计划访问节点 n_ξ^h , 则包含节点服务时间 $s(n_\xi^h)$ 、骑手预计到达时间 $a(n_\xi^h)$ 和骑手预计离开时间 $d(n_\xi^h)$.

由于 $\delta(n, n')$ 在 T 内动态变化, 影响骑手配送时长, 最终导致骑手的实际到达时间和实际离开时间往往与预测不符. 因此在当骑手访问该节点后, 需要将 $a(n_\xi^h)$ 更新为节点的实际到达时间, 并更新实际离开时间. 对于骑手未访问节点, 这些节点组成暂定的待访问路线计划 θ_h^u , 该部分会被取送顺序策略优化改变. 路线计划最后一个节点的时间信息不包括预计离开时间, 只有在路线计划发生更新, 新的节点被插入至该节点后时, 才会更新其预计离开时间.

骑手若在 l_i 前到达餐厅, 则需要等待至备餐完成才能取餐. 骑手如果晚于 f_i 到达客户位置, 则会受到惩罚. 因此, 订单平均超时惩罚计算如下

$$r(I) = \frac{1}{|I|} \sum_{i \in I} [\max(0, a(n_\xi) - f_i)], n_\xi = k_i. \quad (2)$$

若骑手间效益 e_h 相差较大, 则会影响骑手满意度. 因此骑手公平性计算如下

$$r(H) = \sigma^2(e) = \frac{1}{|H|} \sum_{h \in H} (e_h - \bar{e})^2, \quad (3)$$

其中 \bar{e} 是所有骑手的平均收益.

根据骑手的行驶路线, 可以计算出单个骑手完成服务所需的总距离. 加总所有骑手的总距离可得服务所需的总距离, 进一步求出单个订单的平均服务距离, 计算公式如下

$$r(\theta) = \frac{1}{|I|} \sum_{h \in H} \sum_{i=1}^m d(n_{i-1}, n_i), n_i \in N, \quad (4)$$

其中 m 是骑手路线计划中节点的总数(包括起点和终点). n_i 表示路线中第 i 个节点.

综上, 本文优化目标为运营周期 T 内, 订单平均超时惩罚、服务距离、骑手公平性之和最小. 认为每个目标同等重要, 权重都定为1, 并对各指标进行Min-Max归一化处理, 目标函数和模型如下

$$\text{Min } W = r'(I) + r'(H) + r'(\theta), \quad (5)$$

s.t.

$$r'(\Omega) = \frac{r(\Omega) - \min(r(\Omega))}{\max(r(\Omega)) - \min(r(\Omega))}, \Omega = I, H, \theta, \quad (6)$$

$$c_h \leq C, \quad (7)$$

$$s(n_\xi^h) = \begin{cases} l_i, n_\xi \in J \\ f_i, n_\xi \in K, \end{cases} \quad (8)$$

$$l_i = t_i + t_p^i, \quad (9)$$

$$\nu_{nn'} = V^* \delta(n, n'), \quad (10)$$

$$a(n_{\zeta+1}) = d(n_\zeta) + \frac{d(n_\zeta, n_{\zeta+1})}{\nu_{n_\zeta n_{\zeta+1}}}, \quad (11)$$

$$d(s_i) \geq a(k_i) \geq l_i, \quad (12)$$

$$d(s_i) \geq d(k_i), \quad (13)$$

$$x_{hi}, x_{nn'}^h \in \{0, 1\}, \quad (14)$$

模型中目标函数(5)表示整体优化目标. 式(6)为Min-Max归一化处理表达式. 约束(7)保证配送过程中骑手满足当前背包容量不超过容量限制. 式(8)表示骑手配送路线中节点的服务时间取值, 式(9)表示最早取餐时间. 式(10)表示各路段的配送速度. 式(11)表示预计到达节点的时间计算公式. 式(12)保证骑手离开商家时间不能早于最早取餐时间. 式(13)保证对于订单 i 满足先取货再送货. 式(14)中 x_{hi} 、 $x_{nn'}^h$ 分别为将订单 i 分配给 h , 以及 h 在配送过程中经过 nn' , 二者都为0-1决策变量.

3 基于分层深度强化学习的取送货优化算法

为应对即时配送问题的需求波动性、路况不确定性以及实时性要求, 本文使用DRL对环境中复杂路况进行感知, 通过不断试错, 优化自身策略. 此外对问题进行分层求解, 在不同任务下考虑各自约束和优化目标, 提高决策质量, 最后利用全局奖励函数将上下层进行关联, 实现协调调度, 充分发挥DRL可自适应订单、路况随机性优势.

3.1 基于SAC的任务分配算法

在上层子问题中, 调度系统通过滚动调度, 根据当前订单池中订单信息、骑手信息, 以及相关路段路况信息, 将任务分配给合适骑手. DRL将调度优化描述为一个马尔科夫决策过程(Markov Decision Process, MDP), 智能体与环境进行交互, 通过执行动作从环境中获得奖励, 并观察到新的状态(States), 进而让智能体通过和环境的不断交互学习得到使奖励最大化的动作策略. 以下是任务分配问题的MDP组成部分:

- 1) 决策点

为确保订单实时性, 本文在订单出现时刻即进行实时决策, 记第 o 个订单为 I_o , 其下单时间为 t_{I_o} . 由于考虑了背包容量限制, 为避免无效调度, 在无可接单骑手时, 不进行决策, 在有可接单骑手时, 再进行决策, 记该时刻为 t_h . 因此第 o 个决策点的决策时间为 $t_o = \min(t_{I_o}, t_h)$.

2) 状态空间

状态包含了在决策点做出决策所需的必要信息. 本文中状态空间由订单池信息、配送资源(骑手)以及路况状态构成. 具体的组成部分如下:

t_o : 决策的发生时刻.

I_o : 待分配订单的详细信息. 包含订单号、下单时间、商家位置、客户位置、最早取餐时间、最晚柔性送达时间等. 每个订单 I_o 可以表示为 $(i_{I_o}, t_{I_o}, s_{I_o}, k_{I_o}, l_{I_o}, f_{I_o})$.

H_o : 决策点 o 骑手集合的详细信息. $H_o = \{H_o^1, H_o^2, \dots, H_o^m\}$, 其中 H_o^h 表示 h 在 o 时的信息集合, 包含当前接单量、最大接单量、当前所处位置、骑手送完当前订单预计时间(最早服务时间), 当前效益. H_o^h 可以表示为 $(c_{H_o^h}, C_{H_o^h}, n_{H_o^h}, t_{H_o^h}, e_{H_o^h})$.

D_o : 决策点 o 时的距离数据和路况信息 δ , 包括骑手两两间距离, 每个骑手当前位置与商家距离、与顾客距离, 订单中的商家与顾客距离. 综上, 本文将上层状态信息表示为一个四元组 $S^{up} = (t_o, I_o, H_o, D_o)$.

3) 动作

在每个决策点智能体根据策略的指导从动作空间 A 中选择一个动作. 在任务分配问题中, 动作为待分配订单-骑手匹配方案. 本文将决策点 o 采取的动作定义为 a_o , 表示将任务分配给某骑手. 即 $a_o = h, h \in H$.

在任务分配后, 被配单骑手的路径将得到更新, 记为 θ_h^o .

4) 奖励

任务分配决策的奖励基于分配前后目标函数的变化, 着重考虑骑手公平性和平均服务距离. 由于较大的骑手公平性和额外距离成本会降低效果, 奖励取这些因素的相反数, 以促进公平分配并减少额外行驶距离. 模型的奖励设计为

$$R_1 = -(r(\theta_h^o) - r(\theta_h) + r(H^o) - r(H)). \quad (15)$$

本文使用Soft Actor-Critic (SAC)进行任务分配, 由于动作空间是骑手编号集合, 属于离散动作空间, 因此将Actor的输出 π_ϵ 修改为 A 的softmax分布, 选择概率最大的作为动作, 即

$$a_o = \operatorname{argmax}(\operatorname{softmax}(\pi_\epsilon(s_t^{up}))). \quad (16)$$

Critic通过接收 S_t^{up} 和动作分布作为输入, 给出在该状态下, 执行Actor的预期回报, 衡量生成的动作好坏. SAC使用了两个Critic来估计预期回报, 并取这两个估计的最小值作为目标值, 减少了价值估计的过乐观偏差, 从而有效缓解了DQN由于Q值估计偏差引起的学习不稳定性问题^[20,21], 即

$$y_i = r_i + \gamma \min_{j=1,2} Q_{\omega_j^-}(s_{i+1}, a_{i+1}) - \alpha \ln \pi_\epsilon(a_{i+1} | s_{i+1}), a_{i+1} \sim \pi_\epsilon(\cdot | s_{i+1}). \quad (17)$$

基于最大熵原则, SAC引入额外的参数 α 来控制熵项和价值函数之间的权衡, 使得SAC能够在探索与利用之间自动达到良好的平衡^[22], 确保算法既能在环境中充分探索, 又能有效地利用已有知识进行决策. 这种对探索性和稳定性的增强, 使得SAC在更广泛的任务范围内实现了高效学习, 并且能够更好地处理复杂和动态变化的环境挑战.

在Actor网络时, SAC的损失函数不仅包含 Q 值的负期望, 还加入了熵的正则项(式(18)左半部分). 这意味着Actor网络不仅被训练成选择高回报的动作, 同时也被鼓励增加其输出分布的随机性(即增加熵), 以促进更广泛的探索行为. 此外还通过自适应调整 α 使得策略的熵接近一个目标熵值(式(19)). 通过最大化期望回报与预期累积熵, SAC确保了算法不仅能有效地利用已有知识进行决策, 还能够维持足够的探索力度, 避免过早收敛到局部最优解. 这种机制加速了全局最优策略的发现, 使得SAC在复杂和动态环境中表现出更好的

性能、稳定性和适应性,同时减少了陷入较差局部最优的可能性,进而加快学习速度并提高最终策略的质量。

$$L_{\pi}(\epsilon) = \frac{1}{N} \sum_{i=1}^N \left(\alpha \ln \pi_{\epsilon}(a_i | s_i) - \min_{j=1,2} Q_{\omega_j}(s_i, a_i) \right), \quad (18)$$

$$L(\alpha) = \mathbb{E}_{s_t \sim R, a_t \sim \pi(\cdot | s_t)} [-\alpha \ln \pi(a_t | s_t) - \alpha \mathcal{H}]. \quad (19)$$

使用SAC进行任务分配的策略如下所示:

步骤 1 输入仿真时长 T 、目标熵值 \mathcal{H} 、正则项系数 α 、折扣系数 γ, τ 。

步骤 2 随机生成 $\epsilon, \omega_1, \omega_2$, 并初始化Actor网络 $\pi_{\epsilon}(s)$ 和Critic网络 $Q_{\omega_1}(s, a), Q_{\omega_2}(s, a)$ 。

步骤 3 $\omega_1^- \leftarrow \omega_1, \omega_2^- \leftarrow \omega_2$, 初始化目标Critic网络 $Q_{\omega_1^-}(s, a), Q_{\omega_2^-}(s, a)$, 经验回放池 $\mathcal{D} \leftarrow \emptyset$ 。

步骤 4 获取环境初始状态 s_1 。

步骤 5 根据式(16)选出动作 a_t , 执行 a_t , 根据式(15)计算奖励 R_1 , 获取状态 s_{t+1} ; 将 (s_t, a_t, R, s_{t+1}) 放入 \mathcal{D} 中。其中 $R = R_1 + R_2$, 见3.3部分。

步骤 6 判断 \mathcal{D} 中经验数量是否超过最小数量, 若是, 执行步骤7, 若否, 转步骤8。

步骤 7 从 \mathcal{D} 中随机采样 N 个元组 $\{(s_i, a_i, r_i, s_{i+1})\}_{i=1, \dots, N}$; 根据式(17)估计回报 y_i ; 更新Actor网络: 对 $j = 1, 2$, 以最小化损失函数 $L = \frac{1}{N} \sum_{i=1}^N (y_i - Q_{\omega_j}(s_i, a_i))^2$; 根据式(18)计算损失函数 $L_{\pi}(\epsilon)$ 更新Actor网络, 以最大化预期回报和熵; 根据式(19)更新熵正则项系数 α ; 更新目标Critic网络, $\omega_1^- \leftarrow \tau \omega_1 + (1 - \tau) \omega_1^-, \omega_2^- \leftarrow \tau \omega_2 + (1 - \tau) \omega_2^-$ 。

步骤 8 若时间步 $t < T$, 转步骤5; 否则, 算法结束。

3.2 基于PNLNS的路径优化算法

在配送任务中, 单个订单时骑手路线直接明确, 但订单增多后, 需优化取送路线顺序以提高效率。通过在新增订单时重调度, 确保骑手配送路线最优。具体地, 本文采用了指针网络(pointer networks, PN)与大邻域搜索(large neighborhood search, LNS)相结合的方法PNLNS进行路径优化。

首先将由 $n_h, \theta_h^u, \delta(\ell, \ell'), \ell, \ell' \in n_h \cup N_h^u$ 构成的下层状态信息 S_d 输入PN, PN依据所接收信息输出路线序列Seq₁, 再使用LNS优化Seq₁, 将LNS优化解Seq₂和Seq₁求交叉熵损失(cross entropy loss, CE)传给PN, 优化PN参数。最后将Seq₂赋给最终优化路线 $\theta_h^{u'}$, 根据式(20)求解下层奖励, 传递给上层。 $I_h^u, I_h^{u'}$ 表示优化前后路线中的订单情况。执行路径优化得到的奖励 R_2 计算公式如下

$$R_2 = -(r(\theta_h^{u'}) - r(\theta_h^u) + r(I_h^{u'}) - r(I_h^u)). \quad (20)$$

PN作为一种序列到序列(Seq to Seq)模型, 具有注意力机制, 能够感知捕捉到订单间的依赖关系、紧迫性及节点间的距离, 每次给出下一个最佳访问节点, 逐步构建完整的配送路径。此外, PN能够适应任意长度的序列输入, 并且每次生成的下一个节点都是从当前未访问的节点中选出的, 确保了路径中不会重复访问同一个节点, 这赋予了PN良好的扩展性和灵活性。然而, PN在设计时主要关注于通过注意力机制捕捉输入数据之间的依赖关系, 而没有内置对约束的硬性约束, 因此可能生成违反该约束的路径。此外受限于模型结构和训练数据, PN无法充分探索所有可能的路径, 特别是那些需要满足复杂约束的路径。本文使用的PN架构图如图1所示。

LNS算法作为一种经典的路径问题求解方法, 能够在满足多种硬性约束下快速产生大量可行解^[11], 更适合带时间窗的取送货问题求解^[23], 但它不能对环境状态进行感知。将PN与LNS算法相结合, 形成了一种新的优化策略。在这种策略中, PN负责对环境状态进行感知, 包括订单间的依赖关系、紧迫性以及节点间的距离等, 而LNS算法则对PN生成的初步路径进行局部破坏和修复, 以此来增强路径的探索能力。充分利用了PN的序列建模能力和LNS算法的局部搜索优化能力。此外, 将LNS优化后的路径与PN生成的路径之间的

交叉熵损失作为反馈信号,进一步优化PN的参数.使得整个路径优化过程变得更加智能和高效,从而在配送任务中实现更优的配送路线规划.

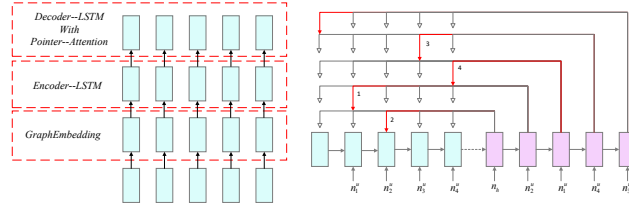


图 1 PN架构图

Fig. 1 Architecture diagram of PN

首先由PN在对路况等随机性进行感知基础上生成初始解,再由LNS增加对潜在最优解的探索,确保路径合理性和效率,最后通过反向传播,提高PN的学习效率和生成解的质量,形成一个完整闭环,提升了路径优化的整体质量. PNLNS策略如下所示:

步骤 1 输入状态信息 S_d, θ_h^u , LNS迭代次数 M , 学习率 ρ .

步骤 2 随机生成指针网络参数 Φ , 初始化最终路径 $\theta_h^{u'} \leftarrow \emptyset$, 初始化PN路径 $\text{Seq}_1 \leftarrow \emptyset$, 初始化LNS路径 $\text{Seq}_2 \leftarrow \emptyset$.

步骤 3 构建距离矩阵 C , 计算原始路线的 $r(\theta_h^u)$ 、 $r(I_h^u)$; PN算法生成初始路线 Seq_1 .

步骤 4 从 Seq_1 随机选出部分节点删除, 放入待插入节点集合; 按照新增成本最小原则, 修复路径, 得到临时解 P .

步骤 5 若 P 满足先取后送约束, 且 $r(\theta_P) + r(I_P) < r(\theta_{\text{Seq}_2}) + r(I_{\text{Seq}_2})$, $\text{Seq}_2 = P$.

步骤 6 若LNS执行次数没有达到迭代次数 M , 转步骤4; 否则, $\theta_h^{u'} \leftarrow \text{Seq}_2$; 计算 Seq_1 与 Seq_2 间的交叉熵损失CE, 计算 R_2 ; 更新参数 $\Phi = \Phi - \rho \nabla_{\Phi} L$; 输出最终路径 $\theta_h^{u'}$ 和 R_2 , 算法结束.

3.3 基于全局奖励的整体优化调度算法

在使用SAC进行任务分配, 使用PNLNS进行路径优化的基础上, 为了实现更加智能和高效的调度决策, 提出了一种创新性的联合训练机制, 该机制将上下层决策通过全局奖励紧密结合起来, 充分发挥各自优势, 并克服了传统启发式算法孤立处理任务分配或路径规划问题的局限性. 具体而言, 借助SAC生成的任务分配方案, 得到骑手行驶路线作为下层路径优化目标, 减小了问题求解复杂度. 在路径优化上, 单独使用指针网络虽然能够对环境变化有所感知, 将信息进行传递, 但其优化性能可能不尽如人意, 结合LNS对指针网络的解进行广泛探索, 寻找更优解, 进一步优化指针网络, 能够各取所长, 实现高效的路径优化求解能力. 最后通过将上下层奖励整合成全局奖励, 系统实现了信号的有效传递, 进一步优化了任务分配策略网络. 在宏观层面, 追求最佳的整体性能指标, 而在微观层面, PNLNS负责进行具体路径的精细化调整, 确保了系统的灵活性、适应性和解决方案的高质量. 这种上下层策略的有机结合, 有效促进了信息的有效传递和共享, 显著提升了系统的整体性能. 基于全局奖励的整体优化调度策略如下所示:

步骤 1 输入仿真时长 T .

步骤 2 初始化待分配订单 $D \leftarrow \emptyset$, 初始化可接单骑手集合 $H \leftarrow \emptyset$, $c_h = 0$.

步骤 3 若新订单 i 出现, 将 i 放入 D ; 若 $c_h < C$, 将 h 放入 H .

步骤 4 若 D 与 H 均不为空, 获取状态 s , 基于SAC选择合适接单骑手 a , 将 s_i, k_i 插入 a 的 θ_a^u ; 计算上层奖励 R_1 , 获取新状态 s' ; 使用PNLNS优化 a 的取送顺序, 得到 $\theta_a^{u'}$, 计算下层奖励 R_2 ; 计算全局奖励 $R = R_1 + R_2$; 将 (s, a, R, s') 放入经验回放池用于策略网络训练; 否则, 骑手按路线进行配送, 到达取货位置 $c_h + 1$, 到达送货位置 $c_h - 1$.

步骤 5 若当前时刻 $t < T$, 转步骤3; 否则, 算法结束.

4 仿真实验及结果分析

4.1 实验数据及仿真环境介绍

本文使用某外卖平台在大连市的历史真实订单数据进行实验, 数据集中包含订单的产生时间, 下单商家位置, 客户位置信息. 经过对历史订单的出现时间分析, 发现两个订单高峰期, 分别是午高峰(10–13)、晚高峰(17–18). 为验证算法在大量订单场景下的抗压能力, 本文选取这两个高峰段订单数据进行实验. 此外为验证算法的适应性, 本文生成了模拟数据集进行实验.

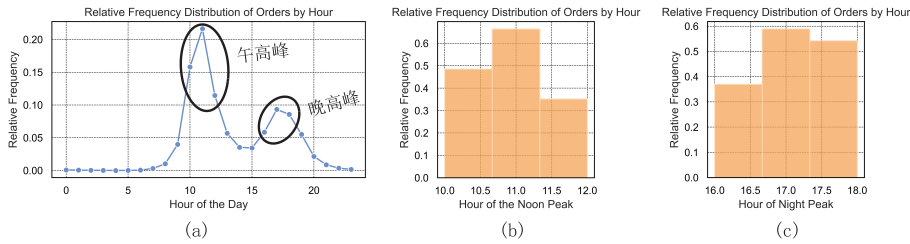


图 2 真实订单分布(a)及模拟午高峰(b)、晚高峰订单(c)分布图

Fig. 2 Distribution of real orders (a) and simulated afternoon peak orders (b) and evening peak orders (c)

本文聚焦于即时配送动态调度问题, 特别是在路况不确定的背景下. 在实验过程中, 我们综合考虑了多种随机性因素, 以更准确地模拟和解决实际问题. 表 1详细列出了这些随机因素及其取值的依据和方法.

表 1 随机性来源表
Table 1 Randomness source table

随机因素	说明	取值分布
订单随机性	订单的出现时间、商家和客户的位置不确定	真实订单: 订单原始数据 模拟订单: 在多天历史数据中采样, 相关信息服从以下经验分布: $t_i \sim F_{t_I}(x), j_i \sim F_{j_I}(x), k_i \sim F_{k_I}(x)$
备餐时间随机性	不同订单所需的准备时间存在差异	$t_p^i \sim N(\mu_1, \lambda_1)$, 其中 $\mu_1=1\ 200, \lambda_1=600$
路况随机性	不同时间、路段的畅通程度不同	$\delta(n, n') \sim F_n(x)$, 其中 $F_{n_1}(x)$ 是基于大连市历史道路畅通程度的数据构建的经验分布函数

本研究通过将大连市路网数据从OpenStreetMap导入SUMO交通仿真软件, 构建了一个真实的交通网络环境, 用于DRL中智能体的交互. 利用Traci接口, 精确控制骑手的配送行为, 模拟真实配送流程, 并实时调整交通状况以模拟路况影响因子对配送的影响. 此外, 通过坐标转换将现实订单信息映射到仿真场景中. 实验过程用到的软硬件配置及环境参数和SAC-PNLNS训练参数设置如表 2所示.

表 2 环境配置及参数设置表
Table 2 The parameters used in Monkey Algorithm

语言/环境	版本/值	参数	值	参数	值
Python	3.10.14	\mathcal{H}	$-\log(1/\hat{h})$	τ	0.005
SUMO	1.15.0	γ	0.98	M	1 000
T	21 600	α	0.01	ρ	0.01
骑手数(\hat{h})	30	C	5	V	25(km/h)

仿真过程图如下所示.

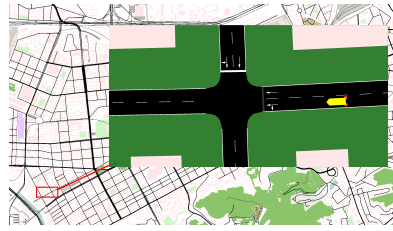


图3 SUMO仿真系统界面与地图区域展示

Fig. 3 SUMO simulation system interface and map area display

4.2 对比实验及性能分析

本节进行了一系列对比实验,验证分层强化学习的综合性能和在状态空间中考虑路况的有效性,并验证上层使用SAC算法的优越性.出于O2O即时配送动态调度的实时性要求(观察到美团、饿了么平台最快为数秒),故选取的比较方法均限定在3 s内须完成一次滚动调度(由于骑手规模影响调度耗时,骑手规模最大时,即最差情况下,纯启发式算法约1秒已收敛,混合式算法约2.8 s, DRL类算法约1.9 s).具体实验如下:

SAC-PNLNS($-\delta$): 在状态空间中去掉从环境中观察得到的路况信息,其他部分保持不变进行调度.

DQN^[24]: 使用DQN算法将任务分配与路径路径优化整合求解.

DQN-PNLNS: 在使用DQN将任务分配给骑手基础上,使用PNLNS对下层路径进行优化.

DDQN-PNLNS: 使用DDQN^[18]将任务分配给骑手,下层使用本文方法进行路径优化.

DP-LNS: 使用距离优先策略,将任务分配给新增距离最少的骑手^[25],再使用LNS对骑手路径进行优化.

SA-LNS^[23]: 经典的两阶段求解带时间窗的取送货问题的启发式算法的改进.第一阶段使用模拟退火算法进行任务分配,第二阶段使用LNS来优化骑手路线.

评价指标如下:

归一化后的平均超时惩罚 $r'(I)$: 见式(2)及式(6).

归一化后的骑手公平性 $r'(H)$: 见式(3)及式(6).

归一化后的订单平均服务距离 $r'(\theta)$: 见式(4)及式(6).

整体优化目标 W : 见式(5).

实验结果如下表所示,其中最优结果进行了加粗,次优使用了下划线展示.

表3 大规模对比实验结果
Table 3 Large-scale comparison of experimental results

数据集	算法	$E(r'(I))(D(r'(I)))$	$E(r'(H))(D(r'(H)))$	$E(r'(\theta))(D(r'(\theta)))$	$E(W)((D(W)))$
真实午高峰	SAC-PNLNS	0.49 (0.02)	0.03 (0.05)	0.18 (0.25)	0.70 (0.28)
	SAC-PNLNS($-\delta$)	0.69 (0.11)	<u>0.04</u> (0.05)	<u>0.30</u> (0.00)	<u>1.03</u> (0.06)
	DDQN-PNLNS	0.82 (0.01)	<u>0.04</u> (0.05)	0.38 (0.39)	1.24 (0.43)
	DQN-PNLNS	0.82 (0.23)	<u>0.04</u> (0.05)	0.34 (0.26)	1.19 (0.07)
	DQN	0.82 (0.26)	<u>0.04</u> (0.05)	0.59 (0.46)	1.45 (0.25)
	DP-LNS	0.26 (0.17)	0.66 (0.03)	0.67 (0.27)	1.58 (0.11)
	SA-LNS	<u>0.36</u> (0.03)	0.98 (0.02)	0.71 (0.20)	2.06 (0.23)
模拟午高峰	SAC-PNLNS	0.42 (0.00)	0.04 (0.05)	0.35 (0.34)	0.80 (0.38)
	SAC-PNLNS($-\delta$)	0.80 (0.27)	<u>0.05</u> (0.06)	0.11 (0.12)	<u>0.95</u> (0.33)
	DDQN-PNLNS	0.82 (0.26)	<u>0.05</u> (0.06)	<u>0.19</u> (0.27)	1.06 (0.08)
	DQN-PNLNS	0.82 (0.21)	0.04 (0.05)	0.27 (0.01)	1.12 (0.15)
	DQN	0.54 (0.13)	0.70 (0.01)	0.62 (0.15)	1.85 (0.27)
	DP-LNS	0.26 (0.16)	0.95 (0.02)	0.46 (0.14)	1.67 (0.17)
	SA-LNS	<u>0.29</u> (0.05)	0.96 (0.03)	0.44 (0.32)	1.69 (0.31)

续表 3
Table 3 Continues

数据集	算法	$E(r'(I))(D(r'(I)))$	$E(r'(H))(D(r'(H)))$	$E(r'(\theta))(D(r'(\theta)))$	$E(W)((D(W)))$
真实晚高峰	SAC-PNLNS	<u>0.20</u> (0.27)	0.02 (0.03)	0.35 (0.50)	0.57 (0.80)
	SAC-PNLNS(- δ)	0.72 (0.09)	0.03 (0.04)	0.39 (0.38)	1.14 (0.51)
	DDQN-PNLNS	0.74 (0.17)	<u>0.03</u> (0.05)	0.44 (0.48)	1.21 (0.36)
	DQN-PNLNS	0.65 (0.15)	<u>0.03</u> (0.04)	<u>0.30</u> (0.36)	<u>0.99</u> (0.55)
	DQN	0.82 (0.26)	<u>0.03</u> (0.03)	0.55 (0.64)	1.39 (0.41)
	DP-LNS	0.19 (0.07)	0.99 (0.01)	0.48 (0.16)	1.66 (0.19)
	SA-LNS	0.23 (0.14)	0.96 (0.02)	0.29 (0.19)	1.48 (0.31)
模拟晚高峰	SAC-PNLNS	0.58 (0.59)	0.01 (0.02)	0.15 (0.17)	0.74 (0.78)
	SAC-PNLNS(- δ)	0.47 (0.67)	0.01 (0.02)	0.33 (0.41)	0.81 (1.09)
	DDQN-PNLNS	0.55 (0.46)	<u>0.02</u> (0.02)	0.20 (0.29)	<u>0.77</u> (0.77)
	DQN-PNLNS	0.38 (0.22)	0.01 (0.01)	0.52 (0.68)	0.91 (0.92)
	DQN	0.73 (0.14)	0.01 (0.02)	0.30 (0.23)	1.05 (0.38)
	DP-LNS	<u>0.41</u> (0.03)	0.97 (0.03)	0.25 (0.08)	1.63 (0.10)
	SA-LNS	0.45 (0.03)	0.97 (0.03)	<u>0.19</u> (0.15)	1.62 (0.15)

由表可知, SAC-PNLNS在四种订单需求分布和随机路况下展现优秀的应变能力, 实现了 $r'(I)$, $r'(H)$ 和 $r'(\theta)$ 三者的均衡, 值远小于其他对比算法; 通过在状态中加入路况信息显著增强了算法性能; 相比单层调度方法(DQN), 使用分层架构能够提高调度效果; 与DQN-PNLNS和DDQN-PNLNS相比, SAC-PNLNS凭借其在目标追求与探索间的良好平衡, 取得了更高的调度效率. 实验表明, 在大规模长期动态调度问题中, DRL方法显著优于两阶段混合启发式方法, 其状态感知和奖励机制能帮助智能体根据环境自适应调整和完美策略, 避免陷入局部最优, 实现整体高收益回报. 此外, 较小的方差表明算法在调度中能够取得探索和利用的均衡, 具有稳定性.

4.3 消融实验及性能分析

为了深入探究各个算法模块对整体性能的具体影响, 本文设计并执行了一系列消融实验. 即去全局奖励信号传递模块(R)、去CE模块、去LNS模块、去PN模块. 为了使实验更具针对性且资源利用更高效, 从数据集中随机抽取了350条订单数据, 并将其产生时间缩放到一小时内, 模拟20名骑手进行配送的场景. 以SAC-PNLNS作为基准算法, 对比变体的性能, 从而准确识别出个模块对整体性能的贡献. 实验结果如下:

表 4 消融实验结果
Table 4 Ablation results

实验编号	算法	$E(r'(I))(D(r'(I)))$	$E(r'(H))(D(r'(H)))$	$E(r'(\theta))(D(r'(\theta)))$	$E(W)((D(W)))$
1	SAC-PNLNS(基准)	0.22 (0.15)	0.19 (0.03)	0.02 (0.02)	0.44 (0.17)
2	- R	0.25 (0.10)	0.86 (0.08)	0.84 (0.13)	1.95 (0.11)
3	- CE	0.45 (0.48)	0.17 (0.23)	0.85 (0.09)	1.46 (0.70)
4	- LNS	0.15 (0.20)	0.86 (0.11)	0.83 (0.15)	1.84 (0.45)
5	- PN	0.23 (0.10)	0.74 (0.13)	0.78 (0.13)	1.75 (0.18)

表4中, 对比1和2, 去除 R 模块后算法性能显著降低. 缺少 R 阻碍了高层与低层之间的有效反馈, 使得各自优化目标产生冲突, $r'(H)$ 和 $r'(\theta)$ 显著恶化, 大幅降低整体优化效果. 对比1和3, 去除CE模块后, $r'(I)$ 和 $r'(\theta)$ 显著增加, 性能下降, 表明相比简单使用LNS对PNS解探索, 引入损失回传机制进一步优化PN模块, 能够提高路径优化的效果. 综合3、4和5, 单独的PN、LNS分别在 $r'(I)$ 和 $r'(\theta)$ 上有较好性能, 但在其他方面存短板. 将PN与LNS结合起来, 能够有效地融合两者的优势, 弥补各自短板, 提升整体性能.

4.4 多场景性能分析

此前分析了不同订单分布下大规模骑手调度, 为了进一步分析订单、骑手数量比例对调度的影响, 设计了三种负载类型下, 12个细分场景, 并使用SAC-PNLNS进行调度. 以下是具体场景设置参数表.

应性, 确保了在资源紧张和任务繁重时仍能维持高效的调度表现. 这些特性使得SAC-PNLNS算法在复杂多变的实际操作中具有极高的应用价值. 此外, 通过及时调整骑手数量, 改善订单-骑手比, 有望进一步提高SAC-PNLNS算法优化性能. 与混合启发式算法相比, SAC-PNLNS在应对不确定性和动态变化方面的表现更优, 进一步证实了其在解决复杂调度问题时的有效性和可靠性.

5 结束语

本文针对即时配送服务中的需求波动性、路况不确定性和实时性要求, 提出了一种综合考虑平台、客户和骑手满意度的优化模型, 并使用SAC-PNLNS 分层DRL方法求解. 该方法结合了DRL与启发式算法的优势, 并通过全局奖励函数实现上下层信号传递, 确保整体优化调度. 最后进行了大量仿真实验, 验证了本算法相较于对比DRL方法和混合启发式算法在性能上的优越性, 以及SAC调度优越性和下层算法各模块的有效性. 多场景实验结果表明, 相较一系列对比算法, SAC-PNLNS能高效应对环境变化和 demand 波动, 保持调度的稳定性与适应性. 此外研究还发现, 通过监测订单量并动态调整骑手数量, 算法不仅能在非高峰期、实现降本增效, 还能在高峰期显著提升调度效果和服务水平, 为即时配送服务提供了兼具成本效益和服务质量的优化调度解决方案. 未来工作中, 我们将探索在调度系统中整合外包骑手资源, 通过动态分配不同来源骑手数量进一步优化调度效率, 以及探索更有效的混合算法策略.

参考文献:

- [1] 马艳芳, 赵媛媛, 周晓阳, 等. 考虑成对取送点的O2O订单配送路径优化. 系统工程学报, 2024, 39(6): 801–820.
Ma Y F, Zhao Y Y, Zhou X Y, et al. Delivery routing optimization problem for O2O orders with paired pick-up and delivery nodes. *Journal of Systems Engineering*, 2024, 39(6): 801–820. (in Chinese)
- [2] Zhang K, Li M, Wang J, et al. A Two-stage Learning-based method for Large-scale on-demand pickup and delivery services with soft time windows. *Transportation Research, Part C: Emerging Technologies*, 2023, 151: 104122.
- [3] 周成昊, 吕博轩, 周翰宇, 等. 以商圈为中心的O2O动态外卖配送路径优化模型与算法. 运筹学学报, 2022, 26(3): 17–30.
Zhou C H, Lü B X, Zhou H Y, et al. Optimization model and algorithm of O2O dynamic take-out delivery route centered on business district. *Journal of Operations Research*, 2022, 26(3): 17–30. (in Chinese)
- [4] Furtado M G S, Munari P, Morabito R. Pickup and delivery problem with time windows: a new compact two-index formulation. *Operations Research Letters*, 2017, 45(4): 334–341.
- [5] Bettinelli A, Cacchiani V, Crainic T G, et al. A branch-and-cut-and-price algorithm for the multi-trip separate pickup and delivery problem with time windows at customers and facilities. *European Journal of Operational Research*, 2019, 279(3): 824–839.
- [6] Aziez I, Côté J F, Coelho L C. Exact algorithms for the multi-pickup and delivery problem with time windows. *European Journal of Operational Research*, 2020, 284(3): 906–919.
- [7] Hou Y, Guo X, Han H, et al. Adaptive ant colony optimization algorithm based on real-time logistics features for instant delivery. *IEEE Transactions on Cybernetics*, 2024, 54(11): 6358–6370.
- [8] Sun P, Veelenturf L P, Hewitt M, et al. Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows. *Transportation Research, Part E: Logistics and Transportation Review*, 2020, 138: 101942.
- [9] Li J, Yang S, Pan W, et al. Meal delivery routing optimization with order allocation strategy based on transfer stations for instant logistics services. *IET Intelligent Transport Systems*, 2022, 16(8): 1108–1126.
- [10] Liang X, Yang H, Wang Z. Rolling optimal scheduling for urban parcel crowdsourced delivery with new order insertion. *Computers & Operations Research*, 2024, 171: 106779.
- [11] 王新玉, 唐加福, 赵志明, 等. 外卖平台在线订单分配及骑手调度优化. 系统工程学报, 2024, 39(5): 724–734.
Wang X Y, Tang J F, ZHAO Z M, et al. Online order assignment and rider scheduling optimization for take-away platform. *Journal of Systems Engineering*, 2024, 39(5): 724–734. (in Chinese)
- [12] Chen X, Yao L, McAuley J, et al. Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowledge-Based Systems*, 2023, 264: 110335.

- [13] Shroff A, Shah B J, Gajjar H. Online food delivery research: A systematic literature review. *International Journal of Contemporary Hospitality Management*, 2022, 34(8): 2852–2883.
- [14] Jiang L, Wang S, Guo B, et al. FairCod: A Fairness-aware Concurrent Dispatch System for Large-Scale Instant Delivery Services//*Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023: 4229–4238.
- [15] 凌 帅, 杨 娟, 孙 鹏, 等. 多目标协同下的即时配送路径优化. *交通运输工程与信息学报*, 2025, 51(9): 328–339.
Ling S, Yang J, Sun P, et al. Real-time distribution route optimization under multi-objective collaboration. *Journal of Transportation Engineering and Information*, 2025, 51(9): 328–339. (in Chinese)
- [16] Jahanshahi H, Bozanta A, Cevik M, et al. A deep reinforcement learning approach for the meal delivery problem. *Knowledge-Based Systems*, 2022, 243: 108489.
- [17] Li J, Xin L, Cao Z, et al. Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(3): 2306–2315.
- [18] Zou G, Tang J, Yilmaz L, et al. Online food ordering delivery strategies based on deep reinforcement learning. *Applied Intelligence*, 2022: 1–13.
- [19] 陈彦如, 刘珂良, 冉茂亮. 基于深度强化学习的外卖即时配送实时优化. *计算机工程*, 2025, 51(9): 328–339.
Chen Y R, Liu K L, Ran M L. Real time optimization of food delivery based on deep reinforcement learning. *Computer Engineering*, 2025, 51(9): 328–339. (in Chinese)
- [20] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor // *Proceedings of the 35th International Conference on Machine Learning Research*, 2018: 1861–1870.
- [21] Wu J, Huang Z, Lv C. Uncertainty-aware model-based reinforcement learning: Methodology and application in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2022, 8(1): 194–203.
- [22] Tang H, Wang A, Xue F, et al. A novel hierarchical soft actor-critic algorithm for multi-logistics robots task allocation. *IEEE Access*, 2021, 9: 42568–42582.
- [23] Bent R, Van Hentenryck P. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 2006, 33(4): 875–893.
- [24] Bozanta A, Cevik M, Kavaklioglu C, et al. Courier routing and assignment for food delivery service using reinforcement learning. *Computers & Industrial Engineering*, 2022, 164: 107871.
- [25] 王新玉, 唐加福, 邵 帅. 多车场带货物权重车辆路径问题邻域搜索算法. *系统工程学报*, 2020, 35(6): 806–815.
Wang X Y, Tang J F, Shao S. Local search algorithm for the multi-depot weighted vehicle routing problem. *Journal of Systems Engineering*, 2020, 35(6): 806–815. (in Chinese)

作者简介:

高 明(1980—), 男, 甘肃白银人, 博士, 教授, 博士生导师, 研究方向: 深度学习与调度优化算法, Email: gm@dufe.edu.cn;

陈明浩(1998—), 男, 安徽六安人, 硕士, 研究方向: 强化学习, Email: cmh98117n@163.com;

唐加福(1965—), 男, 湖南东安人, 博士, 教授, 博士生导师, 研究方向: 运作管理, Email: jftang@mail.neu.edu.cn;

邹广宇(1979—), 男, 辽宁辽阳人, 博士, 副教授, 硕士生导师, 研究方向: 计算机仿真优化, Email: gyzhou@dlut.edu.cn;

许 欣(2001—), 女, 陕西汉中, 硕士, 研究方向: 深度强化学习, Email: 2389968827@qq.com.