控制与决策
Control and Decision

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

周梦, 王境琦, 吴楚格, 夏元清

引用本文:
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.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2025.0893

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

周 梦，王境琦，吴楚格†，夏元清

(北京理工大学 自动化学院，北京 100081)

摘 要: 物流配送效率及其成本优化是制造业供应链管理的核心挑战之一, 相关问题常建模为车辆路径规划问题. 易碎家电等货物在物流运输中无法堆叠, 需在车厢中平铺, 针对这一实际约束, 考虑在传统车辆路径规划模型基础上增加货物的二维装载约束, 形成带有二维装箱约束的车辆路径问题 (2L-CVRP). 该问题包含路径规划与二维装箱两个子问题, 存在强约束、多极组合优化的特性. 传统精确算法及启发式方法在其大规模问题求解上存在耗时长、效率低的局限, 难以应对客户位置、需求即时变化的动态需求. 针对上述快速求解挑战, 设计一种基于强化学习及变邻域搜索协同的知识驱动强化学习求解算法, 优化 2L-CVRP 的车辆行驶距离. 首先, 以车辆行驶距离为奖励设计基于注意力机制与指针网络的 Actor-Critic 强化学习框架, 在此框架下采用多种启发式算法协同处理装箱约束, 改进不可行解, 生成车辆初始路径; 然后, 设计一种高效的问题知识驱动的变邻域搜索策略, 改进端到端网络得到的初始路径序列; 最后, 基于经典 2L-CVRP 测试集验证所提出算法的有效性. 仿真实验表明, 相比经典启发式方法, 所提出算法在小规模实例上车辆行驶距离减少 21.52%, 并更新 50% 的大规模实例最优解. 同时, 所提出算法的求解速度显著优于对比算法, 大规模测例中求解效率优势更加明显, 验证了所提出算法求解 2L-CVRP 的高效性.

关键词: 二维装箱约束下的车辆路径规划问题；强化学习；二维装箱问题；车辆路径规划；组合优化

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2025.0893

引用格式: 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.

Knowledge-driven reinforcement learning method for solving capacitated vehicle routing problem with two-dimensional loading constraints

ZHOU Meng，*WANG Jing-qi*，*WU Chu-ge*†，*XIA Yuan-qing*

(School of Automation，Beijing Institute of Technology，Beijing 100081，China)

Abstract: Logistics distribution efficiency and cost optimization are among the core challenges in manufacturing supply chain management, with related problems often modeled as vehicle routing problems. For fragile goods such as home appliances, which cannot be stacked and must be laid flat during transportation, this practical constraint is incorporated by adding two-dimensional loading constraints to the traditional vehicle routing model, forming the capacitated vehicle routing problem with two-dimensional loading constraints (2L-CVRP). This problem integrates both route planning and two-dimensional packing subproblems, characterized by strong constraints and multi-extreme combinatorial optimization. Traditional exact algorithms and heuristic methods face limitations in solving large-scale instances due to high time consumption and low efficiency, making them inadequate for dynamic demands with real-time changes in customer locations and requirements.To address these rapid-solving challenges, this paper designs a knowledge-driven reinforcement learning algorithm based on the collaboration of reinforcement learning and variable neighborhood search, aiming to optimize the total travel distance in the 2L-CVRP. First, an Actor-Critic reinforcement learning framework based on attention mechanisms and pointer networks is developed, using travel distance as the reward. Within this framework, multiple heuristic algorithms are employed to handle packing constraints and improve infeasible solutions, generating initial vehicle routes. Subsequently, an efficient problem-knowledge-driven variable

收稿日期: 2025-08-29；**录用日期:** 2025-11-26.
基金项目: 国家自然科学基金面上项目 (62573056).
责任编委: 王凌.
†通信作者. E-mail: wucg@bit.edu.cn.

neighborhood search strategy is designed to refine the initial route sequences obtained from the end-to-end network. In terms of simulation experiments, the proposed algorithm is validated on classical 2L-CVRP benchmark sets. Experimental results demonstrate that compared to classical heuristic methods, the proposed algorithm reduces the travel distance by 21.52% on small-scale instances and updates the best-known solutions for 50% of large-scale instances. Moreover, the proposed algorithm significantly outperforms comparative algorithms in solving speed, with advantages becoming more pronounced in large-scale cases, verifying its high efficiency in solving the 2L-CVRP.

Keywords: vehicle routing problem with two-dimensional loading constraints； reinforcement learning； two-dimensional packing problem；vehicle routing problem；combinatorial optimization

0 引 言

目前, 随着电子商务快速发展, 许多物流企业在满足客户需求的同时, 也需要迎接优化运输成本和提高效率的双重挑战. 车辆路径问题 (vehicle routing problem, VRP) 是物流和供应链管理中的核心问题, 其通过规划路径来最小化行驶距离、能源消耗等运输成本[1]. 然而, 传统的 VRP 仅考虑货物的总重量或体积装载约束, 并未考虑货物在车辆内的具体排布情况. 在实际应用中, 例如冷链运输和仓储物流在运送易碎货物时, 货物在物流运输中无法堆叠, 需在车厢中平铺排布以满足二维装载约束[2]. 除传统车辆路径优化外, 还需考虑货物在车辆中的装载方式, 以便合理利用车辆的空间容量, 避免装载不当造成的运输效率降低或货物损坏[3]. 因此, 如何规划运输路线, 设计合理的装箱方案, 从而降低运输成本、提高运输效率具有重要的实际意义.

针对上述实际问题挑战, Iori 等[4] 提出二维装载约束下的车辆路径问题 (two-dimensional loading capacitated vehicle routing problem, 2L-CVRP), 其考虑在给定装载区域的二维平面上装载货物, 同时车辆需要按照一定的路径将货物运输到目的地. 该问题将装载优化与路径优化结合形成 2L-CVRP, 其解决方案能够更贴合实际需求, 提高物流效率.

2L-CVRP 研究工作多为精确算法以及启发式算法. 精确算法利用大量线性不等式对 2L-CVRP 进行整数线性规划建模, 并基于该模型采用分支切割算法最小化车辆行驶距离. 为检验装箱方案可行性, 迭代调用分支定界算法精确计算二维装箱问题, 求解了规模为 35 个客户的实例[4]. Gendreau 等[5] 提出一种禁忌搜索算法, 通过分支定界过程解决问题的装载部分, 该算法在 58 个算例中的 33 个算例更新更优解. 随后, 越来越多的学者设计出不同的启发式算法用于求解 2L-CVRP. Fuellerer 等[6] 对装载部分采用左下方填充、触摸周界算法等几种启发式算法, 整体优化采用蚁群优化算法. 在小规模实例上 ACO 算法达到大量已证明的最优解; 在大规模实例上该算法明显优于已有启发式算法. Zachariadis 等[7] 提出一种融合禁忌搜索及引导局部搜索的元启发式算法, 装载方面使用装箱启发式集合, 路径规划方面使用禁忌搜索方法. 为了加速搜索过程, 该算法减少探索邻域大小, 并采用记忆结构记录装箱可行性信息, 同时在基准实例上更新了几个最优解, 表明通过相关策略引导禁忌搜索过程能够有效加快求解速度. Leung 等[8] 针对 2L-CVRP 提出一种新的元启发式方法, 即扩展引导禁忌搜索, 其对引导禁忌搜索算法进行改进, 帮助诱导禁忌搜索多样化, 使搜索过程覆盖更大的解空间, 从而有效跳出局部最优. 该算法在 Zachariadis 等[7] 开发的装箱算法基础上添加新的装箱启发式算法, 实验结果表明多种装箱算法组合运算能增大算法得到更优解的概率. 随后, Leung 等[9] 提出一种用于求解二维装载异质车队车辆路径问题的带启发式局部搜索的模拟退火算法. 王征等[2] 针对装箱过程使用一种融入启发式知识的深度优先搜索方法, 同时考虑路径优化, 其在 2L-CVRP 算例上有良好的鲁棒性和有效性. 颜瑞等[10]针对多车场带时间窗的 2L-CVRP 进行研究, 在求解车辆路径子问题部分采用量子粒子群算法, 在求解可行装箱方案子问题部分采用引导式局部搜索算法, 在小规模算例上有较好表现. 考虑 2L-CVRP 中的后进先出约束, 尚正阳等[11] 于 2021 年提出 ISA-LOS混合算法, 构建了最低开放空间评估机制, 同时融合基于 Skyline 的空间动态生成方法, 能有效优化装载过程中的空间利用率计算, 其在求解质量和计算效率方面均展现出显著优势.

随着求解问题规模扩大, 已有精确算法及启发式算法在求解 2L-CVRP 时面临挑战. 问题规模扩大, 精确算法计算资源消耗巨大, 时间复杂度呈指数级上升, 难以在有限时间内完成复杂问题的求解. 启发式算法通过迭代逼近最优解, 但无法确保解的绝对最优性, 且用户需要具备一定经验来选择或调整算法参数. 近年来, 强化学习 (reinforcement learning, RL) 通过与环境交互学习优化策略, 在解决复杂组合优化问题方面的应用越来越多[12]. 2016 年, Bello 等[13] 将神经网络与 RL 结合, 提出神经组合优化模

型, 并采用 REINFORCE 算法训练模型参数, 引入 critic 网络构建估值网络, 以降低训练方差. 该算法在 100- 旅行商问题 (traveling salesman problem, TSP) 的求解性能上优于 Christofides 算法和专业求解器. Chen 等[14] 提出针对组合优化问题的 NeuRewriter 框架, 该框架通过迭代不断改进问题的解, 加速问题求解. 针对大规模 TSP 求解, Joshi 等[15] 通过整合监督学习及 RL 进行协同, 针对节点规模为 100 的典型场景展开研究, 其实验数据揭示算法泛化性能受强化学习策略优化机制的制约. Joshi 等[16] 之后对 NCO 模型进行改进, 实现了将 TSP 问题从小范围泛化成大范围 (128 万个节点). Miki 等[17] 针对经典 TSP 问题提出深度学习以及 RL 结合的框架, 采用卷积神经网络将 TSP 问题最优路线作为图像进行卷积学习, 推理阶段利用 EV-贪婪和 EV-2opt 算法, 最终实现最优解的高效输出. 针对覆盖商问题 (covering salesman problem, CSP), Li 等[18] 构建了基于深度强化学习的动态特征编码架构, 采用动态嵌入模型捕捉拓扑结构的动态变化特征, 并运用 REINFORCE 算法更新参数. 实验结果表明, 相较于经典启发式算法, 该智能求解器在保持解质量的前提下可实现计算效率提升约 20 倍, 同时展现出稳定的收敛特性.

然而, 单纯依靠强化学习模型进行端到端求解仍存在收敛速度慢、约束可行性难以保证等问题. 为此, 近年来研究者开始探索强化学习与启发式算法结合的混合框架, 以增强算法的全局搜索效率与可行解质量. Hottung 等[19] 将学习机制嵌入大邻域搜索过程, 生成或选择新的邻域结构, 以增强解空间探索能力. 杨笑笑等[20] 针对 CVRP 设计新的深度混合型邻域搜索模型, 利用深度强化学习模型改进邻域搜索算法, 其模型在 100 规模 CVRP 的优化效果上优于现有 DRL 模型和部分传统算法. Xu 等[21] 在列生成框架中引入 RL 作为超启发式算法, 从而提升列生成的效率与解的整数质量, 在带时间窗的车辆路径规划问题上有较大提升. 这些研究将 RL 模块与启发式模块相结合, 从而实现效率与解质量上的权衡.

针对已有算法求解 2L-CVRP 的局限性, 结合强化学习混合启发式框架在复杂组合优化问题领域展现出的优势, 本文设计一种强化学习及变邻域搜索协同的知识驱动 2L-CVRP 求解算法. 针对车辆路径规划子问题, 所提出算法构建一个端到端的强化学习架构, 采用指针网络与注意力机制实现高效初始路径求解. 问题强约束特性导致端到端模型输出的初始解的可行性无法保证, 算法针对二维装载约束设计装箱检验机制, 分割得到满足所有约束的可行解. 此外, 为了提高解质量, 加入变邻域搜索优化路径方案.

本文结构如下: 第 1 节进行问题描述及其混合整数规划模型介绍; 第 2 节介绍所提出的算法, 包括问题马尔科夫模型、强化学习网络结构及协同装箱启发式规则等; 第 3 节展示所提出算法与经典算法的仿真实验对比结果; 第 4 节给出结论.

1　问题描述与数学模型

1.1　问题描述

2L-CVRP 包含两个子问题: 车辆路径规划问题和二维装箱问题. 该问题要求在满足二维装箱约束的前提下寻找最优的车辆路径规划方案, 最小化车辆行驶距离.

针对 2L-CVRP 有如下假设:

1) 所有车辆都从同一个仓库出发, 在服务完所有客户后返回该仓库;

2) 每辆车的载货容量固定, 在运输过程中不能超过该限制;

3) 每个客户的需求已知, 需要在一次配送中完全满足;

4) 车辆必须从配送中心出发, 连续服务多个客户后直接返回配送中心, 不允许中途返回仓库或更换车辆;

5) 每个客户只能由一辆车服务一次, 不能多次服务或由多辆车共同服务;

6) 装箱方案满足二维装箱约束, 即同一辆车中的货物不能重叠和超出车厢范围.

图 1 为 2L-CVRP 示意图, 其中标序号的圆点为客户节点, 每个客户有若干货物, 如客户 1 共有两个需要运送的货物I_{11}、I_{12}. 车辆 1 从仓库出发, 依次服务客户 1、客户 3、客户 2, 在不违背车辆载重与二维装载约束的前提下将每个客户的所有货物按照一定方案装配在车辆上, 最后回到仓库. 车辆 2 从仓库出发, 依次服务客户 4、客户 5, 最后回到仓库, 完成配送任务.

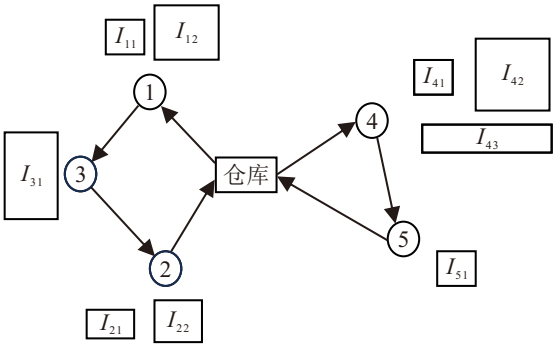


图1　2L-CVRP 示意图

1.2 2L-CVRP 混合整数线性规划模型

根据上述假设与条件说明, 本节首先介绍模型建立过程中用到的参数及其含义, 然后给出 2L-CVRP 的混合整数规划模型.

1.2.1 符号说明

本文用到的符号及其说明如表 1 所示.

表1 符号说明

符号	含义
o	配送中心
N	客户点集合
V	有向图的顶点集, $V=o\bigcup N=\{0,1,\dots,N\}$
A	弧集, $\{A=\|(i,j)\|i,j\in V\}$
G	CVRP有向图, 定义为$G=(V,A)$
M	可用配送车辆集合$M=\{1,2,\dots,k\}$
IT	所有货物集合
IT_i	客户i的货物集合
Q	车辆载重
k	可用配送车辆数
L	车厢装载面长
W	车厢装载面宽
d_{ij}	弧(i,j)的成本, 即点i到点j的欧几里得距离
q_i	第i个客户的需求
d_i	客户i的货物总重
l_{im}	客户i的第m个货物的长
w_{im}	客户i的第m个货物的宽
h_{im}	物品I_{im}左上角距离$(0,0)$的水平距离
v_{im}	物品I_{im}左上角距离$(0,0)$的垂直距离
x_{ij}^k	0-1变量, 取1时表示车辆k经过弧(i,j)
y_i^k	0-1变量, 取1时表示车辆k经过客户点i
Z	目标函数值

1.2.2 混合整数规划模型建立

2L-CVRP 的混合整数规划模型如下:

$$\min Z=\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{M}d_{ij}y_{ij}^k. \tag{1}$$

$$\text{s.t.}\ \sum_{k=1}^{M}\sum_{i=0}^{N}x_{ij}^k=1,\ \forall i,j\in V|i\neq j,\ \forall k\in M; \tag{2}$$

$$\sum_{k=1}^{M}\sum_{i=0}^{N}x_{ij}^k q_i\leqslant Q,\ \forall i\in V,\ \forall k\in M; \tag{3}$$

$$\sum_{k=1}^{M}y_i^k=1,\ \forall i\in V,\ \forall k\in M; \tag{4}$$

$$\sum_{i=1}^{N}x_{ij}^k=y_i^k,\ \forall i\in V,\ \forall k\in M; \tag{5}$$

$$\sum_{j=1}^{N}x_{ij}^k=y_i^k,\ \forall j\in V,\ \forall k\in M; \tag{6}$$

$$\sum_{j=1}^{N}x_{ij}^k=\sum_{j=1}^{N}x_{ji}^k\leqslant 1,i=0,\ \forall k\in M. \tag{7}$$

$\forall i,i'\in\{1,2,\dots,n\},\ m\in\{1,2,\dots,m_i\},$ $m'\in\{1,2,\dots,m_{i'}\},\ i\neq i'$, 需要满足:

$$0\leqslant h_{im}\leqslant W-w_{im}, \tag{8}$$

$$0\leqslant v_{im}\leqslant L-l_{im}, \tag{9}$$

$$h_{im}+w_{im}\leqslant h_{i'm'}, \tag{10}$$

$$v_{im}+l_{im}\leqslant v_{i'm'}. \tag{11}$$

其中: 式 (1) 为目标函数, 即最小化车辆配送总距离; 式 (2) 表示每个客户仅由一辆车服务; 式 (3) 为车辆载重的容量约束, 每条路径上的客户需求量总和不可超过车辆的最大载重限制; 式 (4) 约束保证每个客户都能被车辆服务到; 式 (5) 和 (6) 表示车辆服务时一定有路径连接; 式 (7) 表示所有线路均从同一仓库开始, 服务结束后回到仓库; 式 (8) 和 (9) 表示任何货物装箱时不能超出车厢范围; 式 (10) 和 (11) 表示同一辆车中不同货物之间不能重叠.

2 算法介绍

针对车辆路径规划子问题, 本文构建了一个端到端的强化学习框架, 通过循环神经网络与注意力机制的协同作用有效提升问题的求解效率和解的质量. 针对二维装箱约束子问题, 先后使用 Bottom-Left-Fill、Max-Contact Perimeter 和天际线装箱 3 种装箱检验算法对装箱可行性进行检验. 为了提高解的质量, 在后续路径优化方面引入变邻域搜索 (variable neighborhood search, VNS), 实现在全局搜索与局部搜索间取得较好平衡, 更快找到较优解, 为实际 2L-CVRP 的求解提供高效的解决方案.

2.1 CVRP 子问题求解

针对 CVRP 子问题, 首先建立其马尔科夫决策模型. 在此基础上构建端到端的 Actor-Critic 架构, 通过结合注意力机制的循环神经网络生成路径序列, 并通过策略梯度算法优化模型参数.

2.1.1 马尔可夫决策模型

针对 CVRP 子问题, 首先构建其马尔可夫决策模型. 状态$\{x_t^i=(s_t^i,d_t^i)\}$表示第i个客户在t时刻的状态信息, 包括静态特征信息和动态特征信息. 静态特征为车辆位置二维向量s_t^i, 动态特征为需求状态

$d_t^i=(l_t^i,q_t^i)$, 其中l_t^i为剩余负载, q_t^i为剩余需求. 当前负载对于每个节点均为车辆最大负载容量, 仓库节点的需求为当前车次送出货物和的负数; 当车辆服务某个客户节点后, 该客户需求置为 0.

模型的动作空间由所有尚未访问且车辆有足够容量满足其需求的客户构成. 在 Actor 解码器中通过 mask 掩码机制屏蔽不满足要求的动作, 最终动作由策略网络输出. 例如, 策略网络输出$[0,1,2,0]$表示路径: 仓库→客户 1 →客户 2 →仓库. 通过执行输出动作访问客户, 并更新位置、需求状态. 如果返回仓库, 则车辆负载被重置为最大值. 目标为最小化总距离, 将负路径总距离作为奖励函数.

2.1.2　网络结构

算法采用 Actor-Critic 架构, Actor 网络为编码器-解码器架构, 结合注意力机制与指针网络, 动态捕捉节点的静态特征与动态特征, 生成可行路径[22]. Critic 网络模型为相同的编码器-解码器架构, 解码器部分相比 Actor 模型部分更加简单, 将 3 个卷积层作为其解码器. 下面介绍编码器-解码器具体实现.

输入节点的静态特征s^i与动态特征d^i分别通过静态编码器、动态编码器映射为高维隐藏表示$\bar{s}^i$、$\bar{d}_t^i$, 原始特征被转换为固定维度的嵌入向量$\bar{x}_t^i$.

解码器为 RNN 结构, 基于门控循环单元 (gate-recurrent unit, GRU) 逐步生成路径序列, 结合注意力机制动态调整节点选择概率. CVRP 初始状态为车辆位于仓库, 解码器在每个时间步t输出下一步访问的节点. 初始状态集合X_0中, 任意选择一个输入作为初始点, 用指针y_0指向初始点. 在解码时刻t, y_{t+1}指向X_t中可选的输入, 同时作为下一解码器时刻的输入, 该过程一直持续到满足终止条件; 用Y_t表示t时刻为止解码出的序列$Y_t=\{y_0,y_1,\cdots,y_t\}$. 对于一个解码器, 其输入包括: 1) 当前隐藏状态h_t, 通过 RNN 维护并记录已生成路径的历史信息, 本文在该环节使用 GRU 进行维护记录操作; 2) 嵌入后的输入节点特征$\bar{x}_t^i$: 解码器输出的高维向量.

解码器的输出为当前状态下各节点的未归一化概率分布, 需要通过注意力机制进一步调整. 针对 CVRP, $\bar{x}_t^i=(\bar{s}_i,\bar{d}_t^i)$是输入$i$的嵌入向量, h_t是 GRU 单元在编码步t的记忆状态. h_t拼接编码后的静态特征、编码后的动态特征, 根据式 (12) 计算 attention 得分. attention 得分与编码静态特征加权和, 经过 softmax 计算出每个动作的概率.

在解码器中利用交叉注意力机制, 以解码器隐藏状态h_t为 query, 编码后的静态与动态特征$\bar{s}^i$、$\bar{d}_t^i$为 keys、values, 有

$$u_t^i=\boldsymbol{v}_a^{\mathrm{T}}\tanh(\boldsymbol{W}_a[\bar{x}_t^i;h_t]),\tag{12}$$

$$a_t=\mathrm{softmax}(u_t),\tag{13}$$

$$c_t=\sum_{i=1}^{M}a_t^i\bar{x}_t^i.\tag{14}$$

得到最终的条件概率为

$$\tilde{u}_t^i=v_c^{\mathrm{T}}\tanh(W_c[\bar{x}_t^i;c_t]),\tag{15}$$

$$P(y_{t+1}|Y_t,X_t)=\mathrm{softmax}(\tilde{u}_t^i).\tag{16}$$

通过以上过程, 指针网络将解码器输出映射为节点选择概率, 结合掩码确保动作可行性. 对于指针网络输出的离散概率分布, 模型在训练阶段采用 Categorical 分布采样策略对客户节点采样; 推理阶段选择贪婪解码策略, 将最大概率节点作为下一步选择的客户节点. 图 2 为指针网络推理时从概率分布到输出客户选择的示意图. 在时间步$t=1$时, 概率向量中客户 2 的概率最大, 模型选中客户 2 作为本步输出; 在$t=2$时客户 2 的静态特征作为解码器新的输入, 指针网络得到新的 attention 得分与新的动作选择概率, 直至序列中所有客户被选择.

图 3 为本文算法 Actor 网络模型示意图. 最下方为初始数据输入, 嵌入层将输入映射到高维向量空间. 右侧 RNN 解码器存储解码序列的信息, RNN 隐藏状态和嵌入输入利用注意力机制产生关于下一个输入的概率分布.

在模型中利用掩码机制 mask 辅助更新状态过程. 初始掩码为全 1 矩阵, 每步选择节点后进行更新: 若车辆剩余容量不足, 则屏蔽需求大于剩余容量的

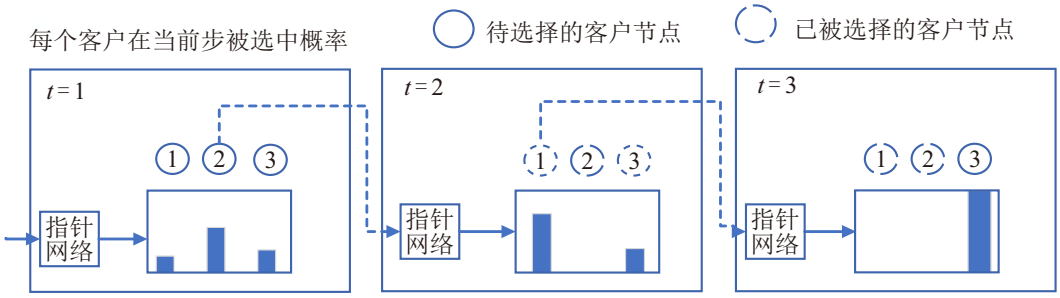


图2　指针网络输出选择客户示意图

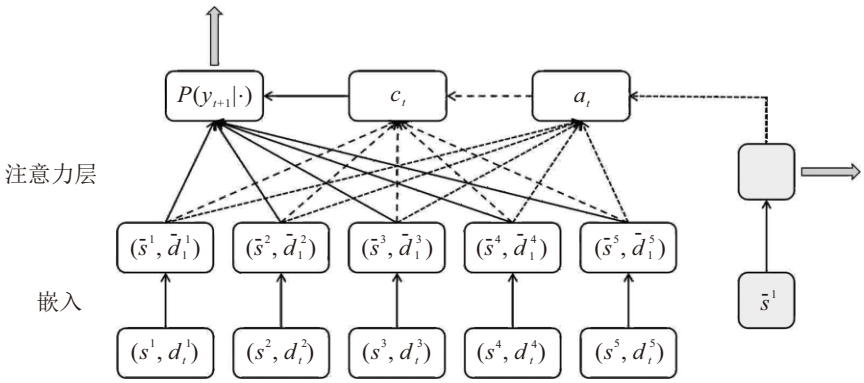


图3 Actor 网络模型

节点; 已访问节点掩码置 0, 避免重复访问, 确保输出解的有效性. Critic 网络部分用于评估当前状态的价值, 输出一个标量估计价值, 其结构与 Actor 网络类似.

2.1.3 损失函数

模型的训练目标为最大化期望奖励 (即最小化路径总距离), 采用策略梯度算法优化参数. 损失函数定义为

$$J(\theta) = -E_{Y\sim\pi_\theta}[R(Y)]. \tag{17}$$

其中: $R(Y)$为路径的奖励 (负路径长度), θ为模型参数.

为降低方差, 引入基线函数$b(s_t)$, 通过 Critic 网络估计实现. 定义优势函数

$$A(Y) = R(Y) - b(s_t). \tag{18}$$

得到模型最终损失函数, 实现参数优化, 即

$$L_{\text{actor}} = -\frac{1}{B}\sum_{i=1}^{B} A_i \cdot \log p(y_i), \tag{19}$$

$$L_{\text{critic}} = -\frac{1}{B}\sum_{i=1}^{B} A_i^2. \tag{20}$$

2.2 二维装箱子问题求解

利用构建的强化学习模型求解 CVRP 子问题, 得到算法初始解. 该初始解不一定为满足二维装载约束的可行解, 需利用装箱算法检验二维装箱子问题, 得到同时满足载重与装箱约束的可行解.

二维装箱子问题的目标是在货物装载不重叠、不越界的前提下, 实现空间利用率的最大化[23]. 本文引入 3 种启发式算法实现快速装箱验证, 分别为 Bottom-Left-Fill 算法[24]、Max-Contact Perimeter 算法[25] 和天际线装箱算法[26].

Bottom-Left-Fill 算法利用经典的左下方启发式规则, 基于贪心策略, 通过双层循环策略优先寻找最低的可用空间位置, 优先填充左下空间以减少碎片化区域.

Max-Contact Perimeter 算法通过最大化货物与已放置区域或车厢边界的接触周长减少空隙, 对每个候选位置计算其接触周长, 选择接触周长最大的位置作为下一个选择的放置位置. 接触周长数学表达式为

$$S(x, y, \theta) = \sum_{e\in\text{edges}} \delta(e)\cdot \text{length}(e), \tag{21}$$

其中$\delta(e) = 1$表示边e与已有区域或车厢边界接触, 否则为 0.

天际线装箱算法中, 天际线表示当前已放置物品的上边界; 新物品的放置位置始终依附于天际线的节点, 从而保证物品不会与已放置物体发生重叠, 充分利用局部空间. 天际线段由左边的端点坐标(x, y)和线段长度决定, 天际线之上的空间被看作未打包区域的子集. 定义一系列位于天际线上的候选点来放置矩形货物, 图 4 为天际线及相应空间的示意图, 选择最低或最左的空间的端点作为放置点, 如空间S_2的端点 3、端点 4. 当物品放置后, 算法会更新全局天际线; 迭代装载直至所有物品装载完成或无可行位置为止. 天际线算法内存消耗少, 计算速度快, 能更灵活地适应物品形状多样性.

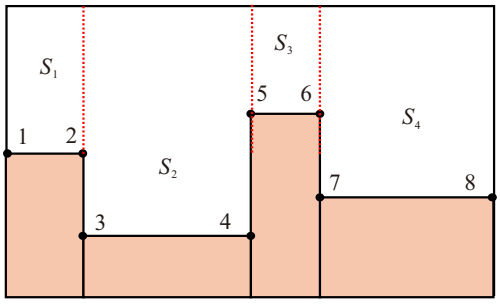


图4 天际线及空间

整体装箱检验算法引入多层检验机制, 逐次调用 Bottom-Left-Fill 算法、Max-Contact Perimeter 算法、天际线装箱算法进行验证. Bottom-Left-Fill 算法适合快速初步检验, Max-Contact Perimeter 算法和天际线算法适合处理复杂形状装箱, 3 种装箱检验可以

兼顾速度和成功率, 节约计算量, 同时避免单一算法失败导致整体失败. 若其中任意一种方法可行, 则判定子路径装箱可行; 若 3 种方法均不可行, 则表明该子路径不满足装箱约束. 表 2 为多层装箱检验算法伪代码.

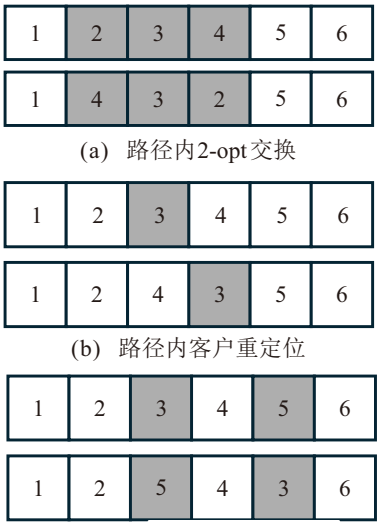
表2　多层装箱检验算法伪代码

算法1	多层装箱检验算法.
输入:	路径route、客户货物集合
输出:	装箱可行性布尔变量True/False、装箱方案$\boldsymbol{P}$
1	初始化:
	从客户货物集合中提取货物, 记录其尺寸信息
	初始化装箱方案$\boldsymbol{P}$为空
2	逐步检验:
	调用Bottom-Left-Fill算法
	if 装箱成功 then
	return True, 对应装箱方案$\boldsymbol{P}$
	else
	调用Max-Contact Perimeter算法
	if 装箱成功 then
	return True, 对应装箱方案$\boldsymbol{P}$
	else
	调用天际线算法
	if 装箱成功 then
	return True, 对应装箱方案$\boldsymbol{P}$
	else
	return False, 该路线不满足装箱约束
3	输出结果:
	return检验结果(True/False)与最终装箱方案$\boldsymbol{P}$

2.3　变邻域搜索算法

为进一步提升解的质量, 引入随机变邻域下降法 (random variable neighborhood descent, RVND) 作为局部搜索策略对初始解进行优化, 提高解的质量. RVND 定义多个邻域结构, 在每一轮迭代中随机打乱邻域结构的顺序, 依次尝试每一个邻域的改进操作, 直到找到更优解. 变邻域搜索的引入可显著提升路径方案的可行性和目标性能[27], 减小车辆行驶距离, 有效优化强化学习方法得到的初始路径. 定义 6 种邻域搜索结构, 分别为路径内 2-opt 交换、路径内客户重定位、路径内客户交换、跨路径 2-opt 交换、跨路径客户重定位、跨路径客户交换, 下面给出每种邻域结构的示意图和说明.

图 5(a) 为路径内 2-opt 交换, 在同一路径内选取两个位置i和j通常要求中间至少存在一个节点, 将两点间的子序列反转. 路径内客户重定位操作在同一路径内, 将某个客户从原位置移除并插入到路径的另一位置. 路径内客户交换操作在同一路径内选取两个不同的客户位置直接交换位置. 跨路径间对应也定义 2-opt 交换、客户重定位、客户交换几种操作, 节点由路径内选取变成不同路径间选取.



(a)　路径内2-opt交换

(b)　路径内客户重定位

(c)　路径内客户交换

图5　邻域结构

RVND 算法步骤如下.

step 1: 以当前解s为初始解, 构建多个邻域结构LS_i.

step 2: 邻域打乱.

在每轮迭代中, 随机打乱邻域的访问顺序.

step 3: 邻域搜索.

依次尝试每个邻域, 得到解s'.

step 4: 解值更新.

采用 first-improvement 策略, 一旦找到更优解s'使得$f(s') < f(s)$, 则更新当前解并重新开始新的邻域序列搜索.

step 5: 停止准则.

算法根据无改进次数停止. 如果达到终止条件, 则结束并输出最优解, 否则返回 step 2.

2.4　整体算法流程

在以上子问题均得到求解的基础上, 本节详细阐述基于强化学习及变邻域搜索协同的知识驱动强化学习求解算法的整体框架及伪代码.

算法框架具体步骤如下 (表 3).

step 1: 初始解生成.

利用构建的强化学习模型求解 CVRP 子问题, 输出结果作为 2L-CVRP 整体算法初始路径方案.

step 2: 变邻域搜索.

对初始路径序列进行变邻域搜索, 多次迭代操作实现对路径解质量的优化.

step 3: 装箱可行性验证.

在该步骤利用装箱多层检验机制分割处理变邻

表3 所提出算法伪代码

算法2	基于强化学习与变邻域搜索的2L-CVRP求解算法.
输入:	客户集合及需求、车辆容量及车厢尺寸、强化学习模型
输出:	满足二位装箱约束的优化路径方案s^*
1	初始化问题实例, 包括客户位置、需求、配送中心
2	初始解生成:
	利用强化学习模型求解CVRP子问题
	得到初始路径方案s_0
3	变邻域搜索:
	对初始路径序列s_0依次执行邻域动作生成候选解s'
4	装箱可行性验证:
	对候选解s'路径分割:
	初始化子路径为空, 依次将客户节点加入子路径;
	若装箱成功, 则继续加入下一个客户; 若装箱失败, 则另起一条新路径
5	迭代优化:
	计算分割后路径方案的总距离$L(s')$
	若$L(s')<$当前最优距离, 则更新最优解$s \leftarrow s'$
	否则跳过该候选解
	重复执行直至达到停止迭代条件
6	输出最终最优解s^*

域搜索的新路径解, 将新路径的节点逐个加入子路径进行装箱可行性检验. 若检验成功则对子路径继续加入客户节点, 直至检验失败, 即表明该节点加入后将违反装箱约束, 该子路径不再执行客户添加操作. 分割示意图见图 6, 新路径将被分割为一条条满足装箱约束以及载重约束的子路径, 最后在每条子路径的起始和终止位置添加配送中心 0, 表示车辆从配送中心出发, 最后回到配送中心.

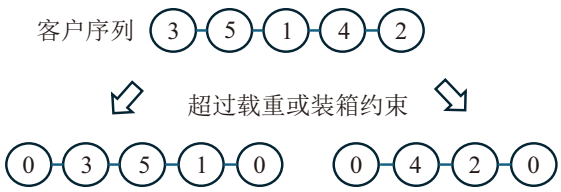


图6 客户序列分割示意图

step 4: 迭代优化.

迭代遍历所有邻域动作, 生成新路径后重复装箱检查与路径分割操作. 对新路径计算其总距离, 如果距离得到优化则将该解加入目前最优值; 否则跳过该动作, 继续搜索其他候选解, 实现路径长度的迭代优化. 图 7 展示了本文提出的 2L-CVRP 混合求解算法的完整流程.

综上, 本文提出的知识驱动混合算法主要由强化学习网络以及变邻域搜索两个主要模块构成. 在整体框架中, 强化学习的策略网络根据问题实例快速生成初始可行解, 评价网络对候选解进行价值估计并为策略更新提供基准, 从而保证训练过程的稳

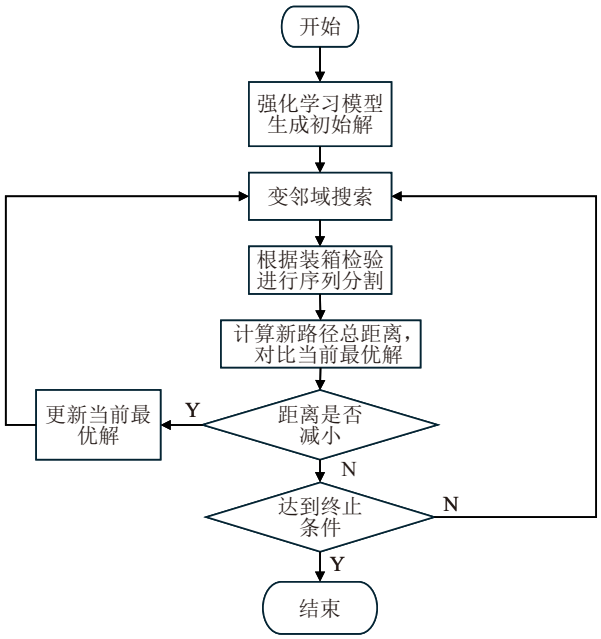


图7 2L-CVRP 混合算法整体流程

定性; 变邻域搜索模块则进一步局部改进解序列. 强化学习网络依赖大量训练数据, 能够学习到具有泛化能力的全局搜索策略; 变邻域搜索利用邻域结构对得到的初始解进行启发式改进. 两个主要模块的结合使得混合算法既具备快速求解能力, 又能获得较优的解质量, 从而在带有二维装箱约束的车辆路径问题中展现出较强的求解性能.

2.5 复杂度分析

本文提出的知识驱动强化学习与变邻域搜索混合算法主要包括强化学习求解子模块与 VND 模块. 算法输入客户节点数n, 隐藏层维度d, 编码器与解码器层数L, 变邻域结构数量K. 强化学习部分的 Actor 网络基于指针网络结构, 需计算节点间的注意力权重, 时间复杂度约为$O(L \cdot n^2 \cdot d)$. Critic 网络为卷积结构, 时间复杂度为$O(n \cdot d)$. 变邻域搜索模块在每轮邻域迭代中遍历节点对, 时间复杂度为$O(K \cdot n^2)$. 装箱检验部分的 Bottom-Left-Fill 算法与 Max-Contact-Perimeter 算法时间复杂度均为$O(n^2)$, Skyline 算法复杂度为$O(n \cdot \log n)$. 模块串行运行, 混合算法整体时间复杂度近似为$O(n^2)$. 空间复杂度方面, 强化学习模块的主要内存开销来源于注意力矩阵与节点嵌入的存储, 其复杂度约为$O(n^2)$; Critic 网络、变邻域搜索及装箱检验部分的空间复杂度分别为$O(n)$量级; 混合算法整体空间复杂度可表示为$O(n^2)$.

3 仿真分析

本实验采用 Python 语言编程, 使用 PyCharm Community Edition 2024.2.4, 计算环境为 11th Gen

Intel(R) Core(TM) i5-11400H@2.70 GHz/16 GB RAM, 操作系统为 windows 11.

3.1　测试数据及算法参数

实验测试采用 Iori 等[4] 提出的 2L-CVRP 实例 (http://www.ordeis.unibo.it/research.html). 对于 Toth 和 Vigo 描述的 36 个 CVRP 实例[28], Iori 等对其扩展, 将客户需求表示为二维、加权和矩形项的集合. 在实验中, 主要算法参数见表 4.

参考 Bello 等[13] 与 Nazari 等[22] 相关工作及小范围预实验, 本文模型训练的学习率设为 5e-4, 保证稳定收敛的同时能较快达到较优性能. 考虑平衡性能与计算成本, 隐藏维度设为 128. 考虑到 GPU 内存限制, Batch size 选用 256. 在模型中引入 0.1 的 Dropout 概率, 用于缓解过拟合问题. 针对规模为 20 的测例, 模型在批大小为 256、20 个 epoch 的训练条件下训练结束需约 3 小时.

3.2　性能实验

为验证本文所提出算法的有效性, 将所提出算法与引导禁忌搜索算法 (guided tabu search, GTS)[7] 在不同规模的 2L-CVRP 实例上进行对比, 实验结果如表 5 所示. 2L-CVRP 混合求解算法对于每个实例独立运行 10 次, 记录最优解与平均解, 同时记录求解时间, 单位为秒 (s). 对于本文混合算法得到的解值与 GTS 解值进行 RPD (相对百分比差异) 计算, 以便直观观察算法性能. 对于实例第 2 类 ～ 第 5 类, 文献 [7] 给出了用 GTS 求解的平均时间, 本文提出的混合算法同样对第 2 类 ～ 第 5 类实例计算平均求解时间, 求解时间平均缩短为第 1 类与第 2 类 ～ 第 5 类求解时间缩短百分比的加权平均值.

表4　参数说明

参数类别	参数名称	数值
算法参数	训练集规模	1 000 000
	Batch size	256
	变邻域搜索最大迭代次数	500
	GTS最大迭代次数	7000
优化器参数	Actor学习率	5e-4
	Critic学习率	5e-4
网络结构参数	编码层数	1
	解码器层数	1
	隐藏维度	128
	Dropout	0.1

表5　本文混合算法与 GTS 算法性能对比

节点数	算例	本文混合算法					GTS		RPD		
		最优解值	求解时间	平均值	平均求解时间/s	标准差	最优解值	求解时间/s	最优解值RPD/%	最优解值求解时间缩短/%	平均求解时间缩短/%
15	0101	**278.73**	1.9	294.34	1.6	11.58	**278.73**	5.2	0	63.46	69.23
	0102	**282.95**	2.9	**295.21**	2.7	14.58	305.92	3.5	−7.51	−30.00	−15.71
	0103	**282.95**	4.5	**290.78**	4.0	9.84	299.70	—	−5.59	—	—
	0104	**294.25**	6.6	301.84	5.6	7.21	296.75	—	−0.84	—	—
	0105	**278.73**	4.2	304.38	3.9	21.96	280.60	—	−0.67	—	—
	0201	**334.96**	1.7	353.87	1.5	19.04	**334.96**	2.4	0	29.16	37.50
	0202	**334.96**	1.7	356.17	1.6	18.03	**334.96**	2.0	0	−22.50	−7.50
	0203	**352.16**	3.4	361.36	2.7	8.97	355.65	—	−0.98	—	—
	0204	345.36	2.6	359.02	2.4	11.75	**342.00**	—	0.98	—	—
	0205	**334.96**	2.1	357.00	1.9	17.96	**334.96**	—	0	—	—
RPD总和/求解时间平均缩短/%									−14.61	−11.74	1.39
20	0301	371.05	2.7	381.95	2.8	7.50	**358.40**	6.8	3.52	60.29	58.82
	0302	**387.70**	5.5	409.93	2.9	18.21	401.81	1.2	−3.51	−216.66	−183.33
	0303	**399.93**	8.9	411.10	3.6	15.02	409.17	—	−2.25	—	—
	0304	**368.56**	6.3	384.02	3.2	8.89	368.56	—	0	—	—
	0305	370.27	5.2	378.12	3.9	6.72	**358.40**	—	3.31	—	—
	0401	433.54	2.9	454.20	2.8	10.58	**430.88**	1.7	0.61	−70.58	−64.70
	0402	**439.89**	2.8	452.81	2.5	9.64	440.94	3.8	−0.23	15.78	24.34
	0403	**430.88**	3.6	455.37	3.1	21.30	446.61	—	−3.52	—	—
	0404	**440.52**	2.9	458.23	2.8	10.29	447.37	—	−1.53	—	—
	0405	439.15	3.5	449.65	3.1	9.58	**430.88**	—	1.91	—	—
RPD总和/求解时间平均缩短/%									−1.69	−81.38	−64.18

表 5 (续)

节点数	算例	本文混合算法					GTS		RPD		
		最优解值	求解时间	平均值	平均求解时间/s	标准差	最优解值	求解时间/s	最优解值RPD/%	最优解值求解时间缩短/%	平均求解时间缩短/%
25	0901	620.25	4.9	660.76	2.5	31.97	**607.65**	1.8	2.07	−61.11	−38.88
	0902	**607.65**	3.2	653.46	3.7	22.75	**607.65**	7.7	0	3.89	33.11
	0903	**607.65**	9.6	635.84	5.8	21.36	622.16	—	−2.33	—	—
	0904	**625.13**	6.2	645.27	4.9	22.66	**625.13**	—	0	—	—
	0905	**607.65**	10.6	656.04	6.2	26.62	**607.65**	—	0	—	—
RPD总和/求解时间平均缩短/%									−0.26	−9.11	18.71
32	1301	2039.67	16.7	2110.64	11.9	71.55	**2006.34**	10	1.66	−37.00	−19.00
	1302	**2622.36**	38.2	2698.55	34.1	80.09	2705.05	68.5	−3.05	−5.76	10.65
	1303	**2477.74**	35.8	2649.95	28.8	145.12	2542.86	—	−2.56	—	—
	1304	**2669.13**	57.3	2808.27	46.6	118.48	2714.69	—	−1.67	—	—
	1305	2451.09	158.5	2701.90	135.3	208.65	**2434.99**	—	0.66	—	—
RPD总和/求解时间平均缩短/%									−4.96	−12.01	4.72
50	1901	535.97	35.4	570.88	31.4	32.15	**524.61**	169.1	2.16	79.06	81.43
	1902	800.91	81.4	827.41	61.1	21.79	**796.87**	570.7	0.50	69.96	112.25
	1903	**811.79**	87.6	834.32	72.2	18.93	816.77	—	−0.60	—	—
	1904	**817.19**	143.6	857.28	125.7	38.58	819.79	—	−0.31	—	—
	1905	683.84	373.1	710.46	340.8	21.13	**674.20**	—	1.42	—	—
RPD总和/求解时间平均缩短/%									3.17	71.78	106.09
75	2101	698.99	96.5	725.75	88.0	15.73	**687.60**	359.1	1.65	73.12	75.49
	2102	1078.46	156.8	1099.61	154.2	20.69	**1076.24**	203.8	0.20	−73.44	63.96
	2103	**1163.95**	295.1	1210.03	272.1	37.95	1191.07	—	−2.27	—	—
	2104	**1018.05**	296.5	1060.72	247.4	51.06	1019.74	—	−0.16	—	—
	2105	941.42	665.5	995.11	468.1	51.47	**914.68**	—	2.92	—	—
	2201	785.48	75.2	805.85	63.8	17.39	**740.66**	1160.9	6.05	93.52	94.50
	2202	1112.33	184.4	1145.00	151.3	25.60	**1088.33**	1584.6	2.20	80.68	105.09
	2203	**1109.21**	240.5	1149.32	218.1	19.15	1110.73	—	−0.13	—	—
	2204	**1113.25**	294.7	1145.08	283.3	20.80	1119.34	—	−0.54	—	—
	2205	991.4	504.9	1014.95	431.5	17.37	**986.02**	—	0.54	—	—
RPD总和/求解时间平均缩短/%									10.46	19.56	84.62
100	2601	831.81	189.6	889.43	172.4	38.53	**819.56**	1743.1	1.49	89.12	90.10
	2602	**1366.97**	340.4	1451.42	285.1	70.58	1387.30	1403.9	−1.46	55.70	103.96
	2603	**1413.00**	562.1	1473.10	535.7	46.50	1436.55	—	−1.63	—	—
	2604	**1467.6**	653.7	1534.68	636.9	44.46	1491.00	—	−1.56	—	—
	2605	**1256.28**	931.1	1348.95	842.1	75.54	1267.68	—	−0.89	—	—
RPD总和/求解时间平均缩短/%									−4.05	62.38	101.19
								总和	−11.95	39.48	252.54

注: GTS数据来自文献[7]等, 标粗表示该测例的最优解.

根据表 5 实验结果, 求解质量方面, 本文混合算法在 64% 算例上更新了 GTS 的最优解, 算例最优解 RPD 总和为−11.95%, 验证了所提出算法性能的优越性. 特别是在问题规模较小的情况下, 如节点数为 15 的情况下, RPD = −14.61%, 相比已有启发式规则性能得到改进. 此外, 在求解效率方面, 混合算法的求解速度显著优于 GTS, 在节点数规模增大时效率优势更明显, 保证与 GTS 同等甚至更优求解精度的情况下大幅提升了求解效率, 尤其适用于需要快速获得高质量解的大规模实际应用场景.

为量化装箱质量指标, 对每个测例统计第 2 类 ∼ 第 5 类的空间利用率均值. 表 6 为测例在混合算

表6　混合算法空间利用率结果

算例	空间利用率/%	算例	空间利用率/%
01	62.69	13	73.08
02	40.84	19	74.45
03	57.42	21	72.48
04	51.17	22	77.72
09	45.95	26	76.72

法最优方案上的空间利用率结果. 在所有测例中, 大多数算例空间利用率可达到 50% 以上, 实现空间利用率的合理利用.

对典型测例 1302 最优方案绘制车辆路线, 其车辆分配及路径规划方案如图 8 所示. 对于算例 1302, 经典算法得到的路径最优值为 2 705.05, 混合算法的最优方案为 2 622.36, 路程长度缩短 3.05%.

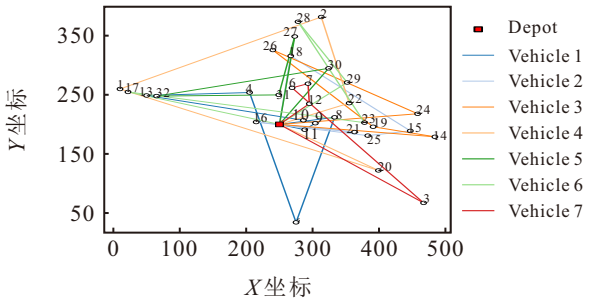


图8　算例 1302 路径方案

3.3　消融实验

为验证强化学习环节为变邻域搜索提供初始解的有效性, 本节设计了消融实验, 对比所提出算法与随机初始化的变邻域搜索算法 (RVND) 的性能差异. 实验数据如表 5 所示, 其中 RPD 为所提出算法相比 RVND 算法的百分比差异, 负值表示混合求解算法解值优于 RVND 解值.

由表 7 可见, 4 类规模下, 所提算法相比 RVND 的最优解 RPD 和平均解 RPD 均小于 0, 表明与随机初始化的变邻域搜索算法相比, 本文所提出强化学习环节引导的混合算法所获得的解质量更优, 强化学习策略通过学习历史经验, 为变邻域搜索提供质量更高的起点, 使其避免在低质量的解空间中执行大量无效搜索, 从而更有效地逼近全局最优解.

表7　强化学习环节消融实验

节点数	算例	RVND算法				RPD			
		最优解值	时间/s	平均值	平均时间/s	最优解值RPD/%	平均值RPD/%	最优解值时间缩短/%	平均时间缩短/%
15	0101	279.6	2.3	294.95	1.9	−0.31	−0.20	17.39	15.78
	0102	282.95	3.8	301.97	2.9	0	−2.23	23.68	6.89
	0103	282.95	4.1	297.43	3.8	0	−2.23	−9.75	−5.26
	0104	294.25	4.5	305.51	3.7	0	−1.20	−46.66	−51.35
	0105	286.35	7.5	289.17	6.4	−2.66	5.26	44.0	39.06
20	0401	458.82	2.4	473.88	2.0	−5.50	−4.15	−20.83	−40.00
	0402	447.55	3.5	487.53	2.7	−1.71	−7.12	20.0	7.40
	0403	439.89	3.9	446.02	3.2	−2.04	2.09	7.69	3.12
	0404	440.52	3.7	467.72	3.4	0	−2.02	21.62	17.64
	0405	453.21	3.9	460.48	3.6	−3.10	−2.35	10.25	13.88
25	0901	642.54	3.3	651.77	3.2	−3.46	1.37	−48.48	−9.37
	0902	642.82	3.6	659.44	3.0	−5.47	−0.90	11.11	−23.33
	0903	642.63	4.3	652.65	3.7	−5.44	−2.57	−123.25	−56.75
	0904	644.88	4.3	647.48	3.9	−3.06	−0.34	−44.18	−25.64
	0905	628.39	5.2	646.50	3.4	−3.30	1.47	−103.84	−82.35
75	2201	805.82	50.0	862.30	40.7	−2.52	−6.54	−50.4	−56.75
	2202	1 139.36	143.3	1 157.36	138.5	−2.37	−1.06	−28.68	−9.24
	2203	1 141.16	195.8	1 156.48	204.5	−2.79	−0.61	−22.82	−6.65
	2204	1 184.23	325.7	1 202.77	297.5	−5.99	−4.79	9.51	4.77
	2205	1 027.18	600.1	1 038.53	580.7	−3.48	−2.27	15.86	25.69
总和/求解时间平均缩短/%						−53.26	−30.44	15.89	11.62

4　结　论

本文围绕二维装箱约束下的车辆路径问题展开研究, 针对传统启发式算法在大规模场景下存在求解效率低、精度低的局限性, 提出了一种数据驱动的基于强化学习与变邻域搜索的算法框架, 实现了路径规划与装箱约束的协同优化. 在路径优化阶段, 基

于 Actor-Critic 架构设计端到端的车辆路径生成策略, 通过注意力机制动态捕捉节点特征; 在装箱优化阶段, 引入 3 种启发式规则进行装箱约束检验; 使用变邻域搜索算法优化路径序列, 提高解的质量. 实验结果表明, 所提出的混合算法在求解质量上与引导禁忌搜索算法相当甚至更优, 同时计算耗时显著降低, 求解效率显著更优. 消融实验进一步验证了强化学习环节能为后续搜索提供优质初始解, 起到引导作用, 避免搜索的盲目性.

在现实物流配送、仓储管理等需快速响应动态变化的应用场景中, 决策者往往对求解速度有极高要求. 本文算法能够灵活权衡解质量和计算效率, 为实时决策系统提供可靠的技术支撑, 有显著现实意义. 然而, 当问题进一步扩展至数百级的更大规模时, 策略网络的计算复杂度将带来较高的时间开销; 在约束条件进一步复杂, 例如引入三维装载或异质货物类型等需求下, 现有启发式装箱算法也不足以完全适配新问题, 表现出一定的局限性. 本研究成果可直接应用于物流配送、储藏管理等场景, 通过优化车辆路径与装载方案, 保障合理空间利用率, 降低运输成本, 为绿色物流以及可持续发展提供技术支撑.

参考文献 (References)

[1] 李国明, 李军华. 基于混合禁忌搜索算法的随机车辆路径问题[J]. 控制与决策, 2021, 36(9): 2161-2169.
(Li G M, Li J H. Stochastic vehicle routing problem based on hybrid tabu search algorithm[J]. Control and Decision, 2021, 36(9): 2161-2169.)

[2] 王征, 胡祥培, 王旭坪. 带二维装箱约束的物流配送车辆路径问题[J]. 系统工程理论与实践, 2011, 31(12): 2328-2341.
(Wang Z, Hu X P, Wang X P. Vehicle routing problem in distribution with two-dimensional loading constraint[J]. Systems Engineering — Theory & Practice, 2011, 31(12): 2328-2341.)

[3] 阳名钢, 陈梦烦, 杨双远, 等. 求解二维装箱问题的强化学习启发式算法[J]. 软件学报, 2021, 32(12): 3684-3697.
(Yang M G, Chen M F, Yang S Y, et al. Reinforcement learning heuristic algorithm for solving the two-dimensional strip packing problem[J]. Journal of Software, 2021, 32(12): 3684-3697.)

[4] Iori M, Salazar-González J J, Vigo D. An exact approach for the vehicle routing problem with two-dimensional loading constraints[J]. Transportation Science, 2007, 41(2): 253-264.

[5] Gendreau M, Iori M, Laporte G, et al. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints[J]. Networks, 2008, 51(1): 4-18.

[6] Fuellerer G, Doerner K F, Hartl R F, et al. Ant colony optimization for the two-dimensional loading vehicle routing problem[J]. Computers & Operations Research, 2009, 36(3): 655-673.

[7] Zachariadis E E, Tarantilis C D, Kiranoudis C T. A guided tabu search for the vehicle routing problem with two-dimensional loading constraints[J]. European Journal of Operational Research, 2009, 195(3): 729-743.

[8] Leung S C H, Zhou X Y, Zhang D F, et al. Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem[J]. Computers & Operations Research, 2011, 38(1): 205-215.

[9] Leung S C H, Zhang Z Z, Zhang D F, et al. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints[J]. European Journal of Operational Research, 2013, 225(2): 199-210.

[10] 颜瑞, 朱晓宁, 张群, 等. 考虑二维装箱约束的多车场带时间窗的车辆路径问题模型及算法研究[J]. 中国管理科学, 2017, 25(7): 67-77.
(Yan R, Zhu X N, Zhang Q, et al. Research on model and algorithm for two-dimensional multi-depots capacitated vehicle routing problem with time window constrain[J]. Chinese Journal of Management Science, 2017, 25(7): 67-77.)

[11] 尚正阳, 顾寄南, 潘家保. 考虑 LIFO 约束的 2L-CVRP 优化[J]. 计算机集成制造系统, 2021, 27(7): 2134-2143.
(Shang Z Y, Gu J N, Pan J B. 2L-CVRP vehicle routing problem with LIFO loading constraint[J]. Computer Integrated Manufacturing Systems, 2021, 27(7): 2134-2143.)

[12] 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.)

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

[14] Chen X, Tian Y. Learning to perform local rewriting for combinatorial optimization[C]. Proceedings of the 33rd Conference on Neural Information Processing Systems. Vancouver: Curran Associates, 2019: 6281-6292.

[15] Joshi C K, Laurent T, Bresson X. On learning paradigms for the travelling salesman problem[J/OL]. 2019, arXiv: 1910.07210.

[16] Joshi C K, Cappart Q, Rousseau L M, et al. Learning the travelling salesperson problem requires rethinking generalization[J]. Constraints, 2022, 27(1): 70-98.

[17] Miki S, Ebara H. Solving traveling salesman problem with image-based classification[C]. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence. Portland, 2020: 1118-1123.

[18] 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.

[19] Hottung A, Tierney K. Neural large neighborhood search for routing problems[J]. Artificial Intelligence, 2022, 313: 103786.

[20] 杨笑笑, 陈智斌. 深度混合型邻域搜索模型求解CVRP问题[J]. 南京大学学报: 自然科学, 2023, 59(6): 1023-1033.
(Yang X X, Chen Z B. Deep hybrid neighborhood search model solves the CVRP[J]. Journal of Nanjing University: Natural Science, 2023, 59(6): 1023-1033.)

[21] Xu K, Shen L, Liu L D. Enhancing column generation by reinforcement learning-based hyper-heuristic for vehicle routing and scheduling problems[J]. Computers & Industrial Engineering, 2025, 206: 111138.

[22] Nazari M, Oroojlooy A, Takáč M, et al. Reinforcement learning for solving the vehicle routing problem[C]. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal, New York: ACM, 2018: 9861-9871.

[23] 张政, 季彬. 考虑随机旅行时间与二维装载约束的越库配送车辆路径优化[J]. 控制与决策, 2023, 38(3): 769-778.
(Zhang Z, Ji B. Optimization for two-dimensional loading constrained vehicle routing problem with cross-docking and stochastic travel time[J]. Control and Decision, 2023, 38(3): 769-778.)

[24] Chazelle B. The bottomn-left bin-packing heuristic: An efficient implementation[J]. IEEE Transactions on Computers, 1983, C-32(8): 697-707.

[25] Lodi A, Martello S, Vigo D. Heuristic and metaheuristic approaches for a class of two-dimensional Bin packing problems[J]. INFORMS Journal on Computing, 1999, 11(4): 345-357.

[26] Wei L J, Oon W C, Zhu W B, et al. A skyline heuristic for the 2D rectangular packing and strip packing problems[J]. European Journal of Operational Research, 2011, 215(2): 337-346.

[27] Souza A L S, Papini M, Penna P H V, et al. A flexible variable neighbourhood search algorithm for different variants of the Electric Vehicle Routing Problem[J]. Computers & Operations Research, 2024, 168: 106713.

[28] Toth E, Vigo D. The vehicle routing problem[M]. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

作者简介

周梦 (2003−), 女, 硕士生, 主要研究方向为智能优化调度, E-mail: 2051574806@qq.com;

王境琦 (2001−), 男, 硕士生, 主要研究方向为车辆路径问题、机器学习与组合优化问题的求解, E-mail: 3220241188@bit.edu.cn;

吴楚格 (1993−), 女, 助理教授, 博士, 硕士生导师, 主要研究方向为智能优化调度、计算系统资源优化, Email: wucg@bit.edu.cn;

夏元清 (1971−), 男, 教授, 博士, 博士生导师, 主要研究方向为多源信息复杂系统的信息处理与控制、云控制与决策理论及其应用、天空地一体化网络环境下多运动体跨域协同控制与智能决策, Email: xia_yuanqing@bit.edu.cn.