国内图书分类号：TP301.6; TP18
国际图书分类号：004.8; 6815

学校代码：10463
密级：□

河南工業大學

硕 士 学 位 论 文

基于深度强化学习的异构车辆路径规划研究

作 者 姓 名　高德玫
指 导 教 师　张闻强　副教授
校外指导教师　王连备　中级工程师

学 位 类 别　电子信息硕士专业学位
专　　业　计算机技术
研 究 方 向　智能信息处理
培 养 单 位　信息科学与工程学院
完 成 时 间　二〇二三年四月

国内图书分类号：TP301.6; TP18
国际图书分类号：004.8; 681.5

密级：公开

基于深度强化学习的异构车辆路径规划研究

学　　号　2020930624

作 者 姓 名　高德玫

指 导 教 师　张闻强　副教授

校外指导教师　王连备　中级工程师

申请学位级别　硕士

专　　业　计算机技术

研 究 方 向　智能信息处理

培 养 单 位　信息科学与工程学院

论文答辩日期　二〇二三年五月二十九

Classified Index: TP301.6; TP18
U.D.C: 004.8; 681.5

Henan University of Technology

Master Degree Thesis

Research on Heterogeneous Vehicle Routing Problem Based on Deep Reinforcement Learning

Student Number: 2020930624

Candidate: Gao Demei

Supervisor: Associate Prof. Zhang Wenqiang

Wang Lianbei

Academic Degree Applied for: Master

Speciality: Computer technology

Affiliations: College of Information Science and Engineering

Dissertation Defense Date: May 29, 2023

摘 要

车辆路径问题是物流运输优化中一个至关重要的问题，它的目标是在满足客户需求的情况下，规划出一条最低成本的车辆路径。现有的基于深度强化学习的解决有容量限制的车辆路径问题的方法本质上是处理同构车队，然而现实中车辆可能是异构的，这使得现有方法效率较低。因此，寻找高性能的算法来提高调度方案的执行效率，成为车辆路径问题中亟待解决的现实需求问题。本研究提出了一种基于注意力机制的深度强化学习算法来求解车辆路径问题，分别实现了带有容量限制的异构车辆、电动汽车路径成本最小化。本文的创新之处如下：

（1）针对异构车辆路径问题，提出了基于注意力机制的深度强化学习方法，目的是最小化车队中车辆的最长行驶时间或总时间。异构车辆的主要特征是容量不同，为了满足异构约束，使用负责异构车辆的选择解码器和一个负责路线构建的节点选择解码器，所选车辆和节点都构成了该步骤的动作。同时采用蒙特卡洛算法进行训练，从而提高模型的求解性能。基于随机生成实例的实验结果表明，本文方法在解决异构车辆路径规划方面优于最先进的深度强化学习方法和大多数传统启发式方法，此外，扩展实验结果表明，该方法也能很好地求解 CVRPLIB 实例，性能令人满意。

（2）针对有容量限制的电动汽车路径问题，提出了端到端的深度强化学习框架，目标使车队的总行驶距离降至最小。同时，开发了一个包含指针网络和图嵌入层的注意力模型，来参数化解决电动汽车路径问题的随机策略。在仅考虑节点信息的框架中，加入图嵌入组件以及全局信息，以综合定义问题的图的局部和整体信息。然后使用奖励函数来评估智能体产生的解决方案，指导智能体进行相应的改进。研究表明，所提出的模型能够有效地解决当前现有方法无法解决的大规模电动汽车路径规划实例。

本文提出的深度强化学习方法与策略，结合了深度学习的感知能力和强化学习的决策能力的优势，可以有效解决有容量限制的车辆路径规划问题，同时对深度强化学习方法解决其它组合优化问题提供了有益的借鉴和参考。

关键词： 深度强化学习算法；异构车辆路径规划；注意力机制；策略梯度

Abstract

The vehicle routing problem is a crucial problem in the optimization of logistics transportation. Its goal is to plan a vehicle route with the lowest cost while meeting the needs of customers. Existing methods based on deep reinforcement learning to solve the capacity-constrained vehicle routing problem essentially deal with homogeneous fleets. However, vehicles may be heterogeneous in reality, which makes existing methods less efficient. The innovations of this paper are as follows:

(1) For the heterogeneous vehicle routing problem, an attention mechanism based deep reinforcement learning method is proposed, with the aim of minimizing the longest travel time or total time of vehicles in the fleet. The main feature of heterogeneous vehicles is their different capacities. To meet the heterogeneous constraints, a selection decoder responsible for heterogeneous vehicles and a node selection decoder responsible for route construction are used. Both the selected vehicles and nodes constitute the actions of this step. At the same time, the Monte Carlo algorithm is used for training to improve the solution performance of the model. Experimental results based on randomly generated examples show that our method outperforms state-of-the-art deep reinforcement learning methods and most traditional heuristic methods in solving heterogeneous vehicle path planning. Solve the CVRPLIB instance with satisfactory performance.

(2) An end-to-end deep reinforcement learning framework is proposed for the capacity-constrained EV routing problem, with the goal of minimizing the total driving distance of the fleet. Meanwhile, an attention model consisting of a pointer network and a graph embedding layer is developed to parameterize a stochastic policy for solving the EV routing problem. In a framework that only considers node information, a graph embedding component is added along with global information to synthesize local and global information of the graph defining the problem. The reward function is then used to evaluate the solutions produced by the agent, guiding the agent to improve accordingly. The study shows that the proposed model can effectively solve large-scale electric vehicle path planning instances that

cannot be solved by current existing methods.

The deep reinforcement learning method and strategy proposed in this paper combines the advantages of deep learning's perception ability and reinforcement learning's decision-making ability, which can effectively solve the problem of vehicle path planning with capacity constraints, and provide deep reinforcement learning methods for solving other combinatorial optimization problems. It is a useful reference and reference.

Key words: Deep reinforcement learning algorithm; Heterogeneous vehicle routing planning; Attention Mechanism; Policy gradient

目　录

1 绪论

1.1 研究背景及意义

车辆路径问题是智能交通系统和运筹学领域的基础课题，在工业领域有着广泛的应用，例如港口、机场和一般仓库的物流。城市物流业备受关注，成为当今社会的一个重要领域，这得益于经济社会的快速发展和交通基础设施的不断完善。2021 年我国快递量首次突破千亿件大关，而在 2022 年，这一数字更是达到了 1105.8 亿件，随着物流业的快速发展，需要更高水平的超大型物流系统快速调度能力，这也促使着调度技术的不断升级[1]。车辆路径规划作为一个众所周知的组合优化问题，是物流配送行业的核心问题。

经典的车辆路径问题被定义如下：在二维空间中有一个仓库节点和若干个客户节点，每个节点的位置和客户需求已知。在满足约束条件的前提下，车辆从仓库节点出发，访问所有客户节点并满足其需求，最终返回仓库节点。给定一定数量的车辆，试图找到一组减少车辆里程和最小化总成本的路线。同时针对优化目标的差异，可以引入各种不同的约束条件，以满足不同类型问题的实际需求。随后，国内外学者提出了许多不同的车辆路径问题变体以满足不同的条件约束，其中有旅行商问题（Travelling Salesman Problem, TSP）和有容量限制的车辆路径问题（Capacitated Vehicle Routing Problem, CVRP）。在不考虑负载的情况下，车辆路径问题假定所有车辆都拥有无限的运载能力，然而旅行商问题则只需要访问每个节点一次，因此这两个问题在负载方面存在差异。在实际生活中，每个车辆的运载能力有限，因此研究有容量限制的车辆路径问题更加实际和重要。有容量限制的车辆路径问题，旨在优化具有容量约束的车队的路线，以满足一组有需求的客户。与同构车辆路径问题中存在多辆相同车辆的假设相比，不同容量（或速度）车辆的设置更符合实际情况，从而得到异构车辆路径问题（Heterogeneous Vehicle Routing Problem, HVRP）[2,3]。根据目标的不同，CVRP 又可分为 min-max 和 min-sum 两类。前一个目标要求车队中车辆的最长（最坏情况）行驶时间（或距离）应尽可能令人满意，因为公平性在许多现实应用中至关重要，后一个目标旨在使整个车队所产生的总行驶时间（或距离）最小化[4]。本文研究了具有 min-max 和 min-sum 目标的 HCVRP 问题，即 MM-HCVRP（Min-Max HCVRP）和 MS-HCVRP（Min-Sum HCVRP）。

目前，为了制定合理高效的物资调度方案，国内外学者研究构建了不同的问题模型以满足多种场景需求，并分别采用精确算法、启发式算法等方法来构造物资调度方案[5]。虽然现有方法在求解时能够取得较好的结果，但精确算法的求解时间会随问题规模的增加呈现快速增长趋势，因此一般仅用于较小规模问题的求解中[6]；而启发式算法的求解性能多受限于手工设计的算法规则，特别是在单配送中心和无时间约束以及道路状况发生变化时所求调度方案在稳定性方面存在不足。近年来，深度强化学习成为迅速崛起的研究方向，研究者致力于推进这一方向的发展。深度强化学习将深度学习与强化学习技术融合在一起，具有广泛适用性。随着深度强化学习的盛行，现如今的国内外学者开始研究其在求解组合优化问题中的应用，其中最具代表性的是解决车辆路径问题。通过使用深度强化学习可以有效地解决车辆路径问题中复杂约束条件和多变的场景问题，从而实现更加高效的路径规划。

随着 HVRP 问题规模的不断加大，约束条件越来越复杂，求解难度不断增加。为了确保配送工作的快速进行，就需要对现有算法进行改进优化，来进一步提升算法探索最优调度方案的能力，从而保证物资及时送达客户。因此，采用深度强化学习方法解决 HVRP 问题，组合了深度学习以及强化学习方法的优势，有助于更好地应对实际应用场景，具有非常重要的现实意义。

1.2 国内外研究现状

1.2.1 车辆路径问题研究现状

自 1959 年被提出来，车辆路径问题（Vehicle Routing Problem, VRP）一直备受国内外学者的关注，成为一个热门话题[7]。随着对 VRP 研究的深入开展，将其与实际场景结合来解决不同问题成为研究的主要方向，为此衍生出多种 VRP 变体问题[8]。但由于它的 NP-hard 特性[9]，随着问题规模的增加，求解最优路径将变得非常困难，因此，研究高性能的算法来寻找最优解成为亟待解决的问题。为此，学者围绕精确算法、近似算法，启发式算法开展了大量研究工作。

精确方法和近似方法

为了解决 HVRP，Yaman[10]提供了六个基于 Miller-Tucker-Zemlin 约束和流量变量的公式。为了获得更好的下界以提高解的质量，通过研究有效的不等式和一些约束来改进公式。Baldacci 和 Mingozzi[11]提出了一种统一的精确方法来解决 HVRP，这有助于

大大减少基于三个边界程序的数学公式中的变量数量，并通过整数规划解决问题。Pessoa 和 Uchoa[12]提出了一种精确的分支削减和定价方法，它在框架中稳健地开发了各种削减，并很好地控制了定价过程的复杂性。通过这样做，所提出的方法可以优化解决具有 75 个客户节点的实例，并在精确方法的行中显示出很大的改进。通过将包括 HVRP 在内的多个车辆路径问题建模为集合划分问题，Baldacci 等人[13]引入了一个精确的框架来解决这些问题。它首先利用双重上升过程产生可接受的解决方案，然后将约束添加到行程中以减少最优性差距，最后应用整数规划来解决问题。Jabali 等人[14]提出了一种连续逼近方法来求解 HVRP，该方法采用混合整数非线性规划并有效地推导其上界和下界。对于具有 min-max 目标的 VRP，Valle 等人[15]引入了分支切割算法来解决选择性车辆路径问题。特别是，精确求解器生成一个可接受的解决方案，该解决方案通过有效的启发式进一步改进以获得更好的性能。Bianchessi 和 Corberán[16]为 min-max 足够接近弧路径问题提供了两个数学公式，并提出了两个精确求解器，包括分支切割算法和分支定价算法，以优化解决该问题，其中开发了专门设计的启发式方法来帮助上限搜索过程。

为了获得最优的问题解，精确算法采用数学方法进行计算，使得所耗费的时间随问题规模增长而大幅度增加，从而降低了算法的求解性能，因此一般将精确算法用在小规模 VRP 问题中。

启发式方法

由于问题的复杂性，通常用启发式方法给出 VRP 的近似解。启发式算法的两个主要类别是构造算法和局部搜索算法[17]。构造算法通过手工制作的规则给出 VRP 的解，牺牲解的质量换取高效率。对于第二类，使用局部搜索的现有方法包括模拟退火[18]、禁忌搜索[19]和大邻域搜索（Large Neighbirhood Search, LNS）[20]。从初始解开始，局部搜索使用不同的搜索算子来搜索更好的解。在现代运筹学中，构造算法通常用于生成初始解[21,22]。然而，局部搜索算法的最终迭代结果很大程度上取决于初始解，特别是对于大规模的问题[23]，因为局部搜索算法的搜索空间在初始解附近。因此，不合适的初始解可能导致大量的计算时间和局部最优。具有 min-sum 目标的车辆路径问题中的异构车队首先被 Golden 等人[2]以无限数量引入，其中应用了一些启发式方法和技术（例如 or-opt 算子）来计算下限并生成接近最优的解决方案。基于这个下界过程，Gheysens 等人[24]中利用启发式方法来解决具有无限车队的 HVRP，该方法首先选择车辆组合，

然后将问题转换为所选车辆的典型 VRP。基于进化方法，Ochi 和 Vianna[25]为 HVRP 开发了一种混合启发式方法，它将并行通用模型和分散搜索与额外设计的分解过程相结合，并提供了有希望的结果。与考虑无限数量的具有不同容量和成本的车辆相比，固定数量的异构车辆更为现实，其目标是最小化总旅行成本和车辆数量，后者作为次要目标。此设置中的 HVRP 很少由精确求解器求解，因此本文重点回顾启发式求解器。Prins[26]采用了一种建设性的启发式方法，通过在容量约束下将部分行程反复合并为一个完整的行程来解决具有固定车队的 HVRP，这也适用于多行程的情况。Vidal 等人[27]提出了一种统一的混合通用搜索方法来解决包括 HVRP 在内的一类车辆路径问题。所提出的方法设计了一个基于问题特定组件的统一局部搜索和一个多样性增强过程，以有效地增加搜索探索。最近，Feng 等人[28]引入了一种进化多任务算法来解决具有时间窗口和偶尔驱动程序的 HVRP，它很好地实现了多任务和高效搜索之间的平衡。

虽然上述开展的启发式算法研究在解决 VRP 时具有较好的求解性能，但由于其性能受到手工设计的局部搜索规则影响，在很大程度上依赖于人类经验和领域知识，对较大规模问题求解时无法对其解空间进行全部搜索，导致所求解与最优解间的偏差难以衡量，因此算法的求解性能还有待进一步提高。

与传统汽车的行驶里程相比，电动汽车的行驶里程较短。因此，它们更有可能在短距离或城市地区使用，在这些地方，它们比传统车辆更有效，因为它们的驾驶速度更低，噪音更低，成本更低[29]。Yilmaz 和 Kalayci[30]开发了一种考虑同时取货和送货的改进 Clarck&Wight 节约算法来构建初始解决方案。然后，提出了总共 12 种解决方案，包括 5 个可变邻域搜索变体和 3 个邻域修改步骤，以改进初始解决方案并在合理的时间内找到最佳解决方案。吴廷映等人[31]设计了改进的自适应大邻域搜索求解考虑载重影响耗电率的电动车路径问题。其设计了多种基于模型特性的破坏算子和修复算子，并引入了禁忌搜索算法的思想。该方法在小规模算例上与 CPLEX 相比具有效率优势，并且在大规模算例上能够稳定运行。针对漫游配送车辆路径问题(Vehicle Routing Problem with Roaming Delivery Location, VRPRDL)中存在的交付计划可能会在计划时间表的执行过程中发生变化这一问题，Ozbaygin 和 Savelsbergh[32]提出了一种分支定价算法来重新优化车辆路线和交付地点。Duman 和 Taş等人[33]开发了基于分支定价切割的精确和启发式方法来解决同一问题，并提出了新的最佳解决方案。然而，随着实例规模的增加，解决方案的质量和效率都会下降。另外，这两种算法都依赖 CEVRP 的结

构和假设，很难推广到其他变体。当平均路线长度较短时，电动汽车更加实用，因为它们可以在返回仓库时进行充电，从而避免了需要在路上寻找充电站的麻烦。由于距离足够短，电动汽车已经被 DHL 和 FedEx 等公司用于轻型货物的最后一英里递送。例如，Ahmadi 等人[34]研究了车辆动力学参数在电动汽车路径的能源模型中的重要性，特别是在提货和交货问题中。Grandinetti 等人[35]开发了一个多目标混合整数线性模型，用于最小化总行驶距离、所用电动车的总成本和未满足时间窗口的总惩罚成本。为了进一步加速搜索过程，Manchanda 等人[36]提出了一个图卷积网络来修剪不良节点并学习良好节点的嵌入，然后将其输入 Li 等人[37]的模型以产生解决方案集。此外，Barrett 等人[38]提出了探索性的深度 Q 网络（Deep Q-Network, DQN），允许算法修改它之前的行动，以便更全面地探索解决方案的空间。

此外，为了将指针网络推广到难以获得实例-解决方案对的组合优化问题，Bello 等人[39]使用策略梯度方法来训练指针网络。指针网络能够有效地找到多达 100 个节点的 TSP 实例的近似最优解。Nazari 等人[40]进一步将这种方法推广到节点状态在解码过程中发生变化的 VRP。考虑到节点的顺序并不能为 VRP 求解器提供任何额外的信息，他们用节点信息的逐元投影取代指针网络中的循环神经网络编码器，从而加速了模型的实现。尽管经过多年的尝试和研究，已经出现了很多解决 VRP 问题的方法，但仍存在一些挑战，比如现有算法的速度和最终解决方案的质量尚未达到理想水平。此外，不同研究人员在不同的背景下关注 VRP，这会导致条件限制的差异，由此导致条件约束的差异，因此某些方法可能仅适用于特定情况。

另外，基于深度强化学习的方法解决车辆路径问题已经成为一种趋势，并且仍然还有很多方法未被探索，因此关于 VRP 的研究非常重要且仍在进行中。

1.2.2 深度强化学习方法研究现状

鉴于深度神经网络在计算机视觉[41]和自然语言处理[42,43]方面取得的显著成功，众所周知，深度模型，尤其是 Transformer 架构，在没有太多人工指导的情况下可以显着优于手工制作的模型[41]。受这种优势的启发，越来越多的人开始探索神经启发式算法来解决诸如 TSP 和 CVRP 等路径问题[44]，它们利用神经模型自动学习传统启发式算法中的搜索规则。通过进一步集成诸如注意力机制等先进的深度学习架构来指导选择，与传统的启发式方法相比，这些基于深度强化学习的方法能够有效地生成质量更高且计算时间更短的解决方案。针对组合优化问题，提出了一种改进传统精确/近似算法的

深度强化学习研究。他们使用机器学习模型来改进经典的精确算法：分支和边界的节点选择和变量选择策略。为了在局部搜索过程中寻找最优解，研究者通常需要手动设计各种启发式规则来构造和搜索解[45]。

基于 Transformer 还有更多的改进，为了进一步提高大邻域搜索的效率和准确性，Falkner 等人[46]提出了一种新的方法，采用修复和破坏算子，通过局部搜索和维护少量候选解来实现搜索的扩展。基于 Falkner 等人的工作，Ma 等人[44]在训练过程中，采用课程学习来提高采样速率、加速收敛速度和减小方差，从而提高模型在求解 VRP 问题时的泛化性能。结果表明该方法优于现有的邻域搜索求解器，并且具有更优的泛化性能。Google 团队[47]提出了由注意力机制和多层感知机组成的网络结构“Transformer”，通过 Transformer 的多头注意力机制，可以提取问题的深层次特征信息。该机制能够从不同的角度和维度注意到子空间的信息，使得节点可以通过多个通道传递相关信息，从而实现并行计算。因此来自编码的节点嵌入可以学习图的上下文中关于节点的信息。Kool 等人[48]提出了一种有效的模型和训练方法，以改进上述基于学习的启发式求解路径问题。通过用注意力机制层代替递归网络来减少节点输入顺序的影响，应用 RL 算法训练模型。

与 Bello 等人[39]的研究不同，该模型的独特之处在于利用引入贪婪策略得到的解作为基线，从而有效提高了模型的收敛速度。Kool 等人将监督学习（Supervised Learning, SL）和 RL 技术结合起来，对 100 个节点的 TSP 进行了训练，与 Joshi 等人[49]的实验设置相匹配，这一方法明显提升了模型的准确率。在 Kool 等人模型中，节点特征通过嵌入方式进行编码，该嵌入在整个过程中是固定的，不会随时间推移而变化。而问题实例的状态应根据模型在不同的构造步骤所做的决定而改变，节点特征应该相应地更新。因此，Peng 等人[50]提出了一种动态注意力模型，采用动态编码器-解码器结构，不同于其它模型的是，它能够在探索节点时自适应地调整，同时在不同的构造步骤中，能够有效地利用隐藏的结构信息。这一模型的应用具有极大的灵活性和有效性，能够为相关领域的研究和实践带来新的思路和方法。相较于 Kool 等人提出的方法，该模型在图的上下文中动态地描述每个节点，从而更加有效地探索和利用隐藏的结构信息，尤其是在不同的构造步骤中，该模型能够展现出更高的灵活性和实用性，为相关领域的研究和事件提供了更为优秀的解决方案。

虽然深度强化学习方法可以直接输出问题的解，但是与专业求解器相比，其优化

效果仍有待提高，需要进一步地改进和优化。传统的组合优化问题通常采用局部搜索方法进行求解，然而为了使搜索规则更具搜索能力，研究人员开始探索使用深度强化学习方法来自动学习局部搜索算法启发式规则，以取代传统的手工设计规则。这种方法能够为搜索算法的改进提供更好的思路和方法，从而使其具有更好的搜索性能。为了解决路径问题并提高算法的性能和效率，Wu 等人[51]提出了一种新的深度强化学习框架。该框架采用强化学习公式改进启发式算法，策略网络节点由节点嵌入和节点选择两部分组成，共同指导下一个解决方案的选择。通过使用 AC 算法训练策略网络和淋雨搜索改进初始解，框架不断提高解的质量。最终，基于自注意力的框架参数化策略进一步优化算法性能。将模型应用于求解 TSP 和 CVRP 的实验结果表明，相比传统的手工规则，改进的启发式算法学习到的策略比现有基于线性规划的求解方法表现更优，这些策略更为有效，而且可以通过简单的策略集成进一步提升求解能力。

多个领域都已经展现了深度强化学习的卓越表现，充分表明了深度强化学习的无限潜力。该领域的研究正在得到越来越多的关注和投入，未来其发展前景将更加广阔。

1.3 存在问题分析

由上述国内外研究现状可知，由于 VRP 的 NP-hard 特性，使得车辆路径规划问题更为复杂。一方面，现有的大多数神经启发式算法都专注于解决两个基本的车辆路径问题，即 TSP 和 CVRP，而很少研究更复杂和实用的约束。另一方面，考虑到随着问题的扩大，车辆路径问题的搜索空间呈指数增长，传统的编码器-解码器结构在当前神经构造方法的有效和多样化探索方面受到限制。

（1）研究异构车队文献较少

相比同构 CVRP 中多个相同车辆的假设，不同容量（或速度）的车辆的设置更符合现实世界的实践，从而导致异构车辆路径问题[2,3]。根据目标，CVRP 也可以分别分为 min-max 和 min-sum。前者要求车队中车辆的最长（最坏情况）行驶时间（或距离）应尽可能令人满意，因为公平在许多实际应用中至关重要，而后者旨在最小化整个车队所发生的总行驶时间（或距离）。现有的神经工作只专注于解决车辆具有相同特征（例如容量和速度限制）的同构 CVRP。

对于这个问题，分配了多辆汽车来服务客户，每辆汽车通过服务一部分客户来完成一次旅行，整个车队为所有客户服务。在这种情况下，通过分配多个相同的车辆来

构建解决方案（具有多个旅行）可以被视为在这些神经工作中多次重复分配单个车辆。这种策略的好处是该策略只需要选择下一个要访问的节点，而无需考虑应该将哪个车辆链接到所选节点。然而，这种策略不能直接解决 HVRP，因为从异构车队中选择车辆很重要，应该在策略网络中加以考虑。

为了简单地应用这些神经工作来解决 HVRP，需要一些策略来使策略能够从异构车队中选择车辆。例如，该策略可以从车队中随机选择一辆车，以及轮流选择一辆车。尽管能够解决 HVRP，但考虑到以下问题，这种适应的神经工作并不有效：

1）策略网络无法自动感知和捕捉异构车辆的差异，鉴于 HVRP 的复杂性，可能无法产生高质量的解决方案；

2）选择车辆也很重要，类似于在 HVRP 中选择节点，在策略中没有明确考虑前者。

（2）神经构造方法中低效的编码器-解码器结构

现有的神经启发式通常通过深度模型利用编码器-解码器结构来参数化概率分布以对解决方案进行采样。以高级强化学习或监督学习的方式进一步训练，编码器-解码器结构在解决 VRP 问题方面表现相当好，特别是在学习神经构造启发式时，它依次决定要访问的下一个节点。给定解码器产生的概率分布，可以对多个解进行采样，并检索出最好的一个作为最终输出。

然而，由于搜索空间可能会随着问题的扩大而呈指数增长，因此有效和多样化的探索对于在有限的计算时间内找到高质量的解决方案至关重要。为此，编码器-解码器结构对于神经构造方法并不是最优的。具体来说，它有以下两个限制：

1）对特定问题的特征进行编码后，在整个采样过程中，从编码器学习到的特征嵌入是固定的，这缩小了搜索范围，忽略了采样历史解的影响；

2）由于固定的特征嵌入，解码器中生成的概率分布是确定性的。尽管采样了多个解决方案，但由于分布不变，它们中的大多数可能本质上是相同的，这可能会严重损害搜索多样性。

在现实世界中，很多问题的状态空间和动作空间都非常大，这给强化学习带来挑战。高维空间会导致维度灾难和计算复杂度的增加。对这个问题，本文使用函数逼近方法，如深度神经网络来处理高维空间。另外，使用基于样本的方法，如蒙特卡洛树搜索，可以在大型动作空间中进行高效搜索。

1.4 本文主要研究内容

针对国内外现阶段对车辆路径规划研究中存在的问题，本文采用机器学习技术中的深度强化学习算法，深入分析算法特征，并对算法进行改进优化以提高其求解性能，并分别用于解决不同约束条件下的物资应急调度问题。主要研究内容如下：

（1）开展异构车辆路径规划问题研究

针对异构车辆不同容量限制问题，提出一种基于深度强化学习的神经构造启发式方法，该方法结合了一个负责异构车辆的选择解码器和一个负责路线构建的节点选择解码器。并采用 REINFORCE 算法进行训练，从而提高模型的求解性能。

（2）电动汽车路径规划分析与优化研究

为了对电动汽车的运营进行建模，提出一个端到端的深度强化学习框架来解决。开发了一个结合了指针网络和图嵌入层的注意力模型，为解决 CEVRP 的随机策略提供参数。然后使用 REINFORCE 策略梯度与 rollout 基线来训练该模型。

1.5 章节安排

本文的章节安排如下：

第一章，绪论。本章首先介绍了深度强化学习以及 VRP 的研究背景及意义，然后总结了国内外相关工作的研究进展，总结了现有方法存在的问题，阐述本文的主要内容和技术路线，并介绍了本文的章节结构。

第二章，深度强化学习算法理论介绍。在本章，简单介绍了深度学习、强化学习以及深度强化学习的相关理论基础和方法。

第三章，利用深度强化学习解决异构车辆路径问题。提出了一种基于注意力机制的深度强化学习方法，其中车辆选择解码器负责异构车队约束，节点选择解码器负责路线构建，该解码器通过在每一步自动选择车辆和该车辆的节点来学习构建解决方案。然后使用具有 rollout 基线的策略梯度来训练模型。实验结果表明，所提出的方法优于最先进的深度强化学习方法和大多数传统启发式方法。

第四章，利用深度强化学习解决带有容量限制的电动汽车路径规划问题。提出了一个端到端的深度强化学习框架来解决电动汽车路径问题。同时，开发了一个包含指针网络和图嵌入层的注意力模型来参数化解决电动汽车路径问题的随机策略。然后使用具有 rollout 基线的策略梯度来训练模型。实验结果表明，所提出的模型能够有效地

解决当前现有方法无法解决的大规模 CEVRP 实例。

第五章，总结与展望。本章对全文车辆路径求解算法及实验结果进行总结，提出可能存在的问题并对下一步工作设想研究方向进行展望。

2 深度强化学习

2.1 深度学习简介

近年来，深度学习（Deep Learning, DL）已经成为机器学习领域中一个非常重要的研究热点，它可以处理更加复杂和大规模的数据，并且可以自动地从数据中提取出更加高层次的特征表示，从而实现对数据的高效处理和准确预测。因此 DL 方法侧重于对事物的感知和表达[52]。

一般来说，DL 模型是由多个非线性单元组成的层级结构。这些层级结构将较低层的输出作为更高层的输入，从大量的训练数据中自动地学习抽象特征表示，并发现数据的分布式特征[53]。尽管如此，由于计算能力不足、训练数据缺乏等原因，使其长期以来未能取得实质性突破。Hinton 等人[54]提出了一种训练深层神经网络的基本原则：先用非监督学习对网络逐层进行贪婪的预训练，然后再结合监督学习对整个网络进行微调以获得更好的性能。这种预训练的方式为深度神经网络提供了较为理想的初始参数，从而能够有效降低深度神经网络的优化难度。此后几年，多种 DL 模型陆续被提出。

作为 DL 技术中备受瞩目的核心技术之一，注意力机制发挥着重要作用，被广泛应用于不同类型的 DL 任务中并产生了多种模型，如 SANet[55]、VSG-Net[56]、ECA-Net[57]等，注意力机制的结构如图 2-1 所示。

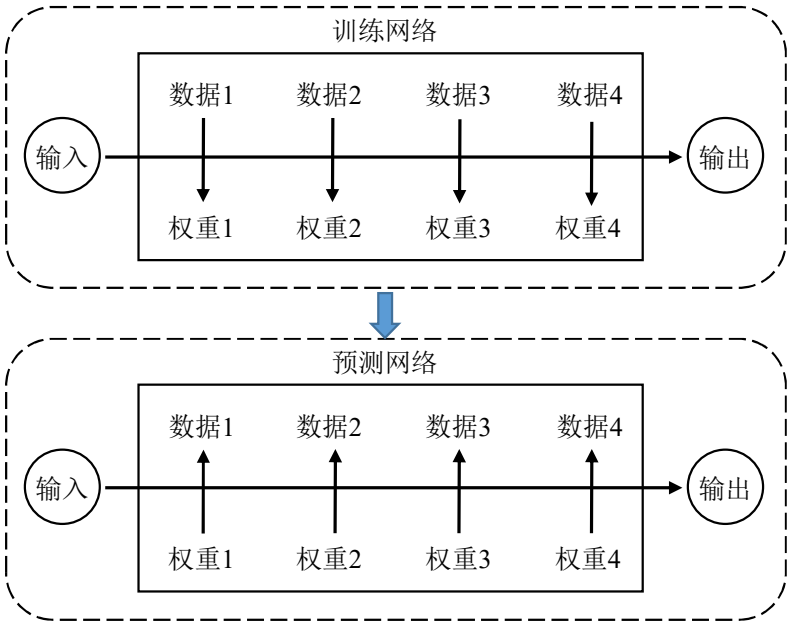

图 2-1 注意力机制结构图

多头注意力（Multi-Head Attention, MHA）机制是目前常用的多种注意力机制之一，能够并行地选择输入信息中的多条信息，该方法通过使用不同的注意力层，关注输入信息的不同部分，从而得到更加精确的输出结果。这些不同的注意力层的结果将被拼接起来，并通过线性变换进行处理，以便最终得到最优的输出结果。基于多头注意力机制的 Encoder-Decoder 模型是在 Encoder-Decoder 框架中使用 MHA，通过集成多个相同注意力的不同结果，获得提升融合后表征能力更强的节点嵌入。在解决 VRP 时，其 Encoder 用于生成中间节点嵌入，并通过 Decoder 依次输出访问节点。

2.2 强化学习简介

2.2.1 强化学习原理概述

强化学习（Reinforcement Learning, RL）是人工智能领域的一个重要分支，目标是通过与环境交互，不断累积奖励（Reward），通过不断尝试和错误来学习正确的决策，以提高智能体的决策能力。与其它机器学习方法不同的是，RL 不需要监督信号来进行学习，而是依赖智能体在环境中的反馈回报信号。智能体根据反馈回报信号对其状态和动作进行更正，逐步实现奖励的最大化。这种自我学习能力使得 RL 具有很强的自主性和灵活性，能够在不同环境中学习和适应。图 2-2 总结了一些已经应用到 VRP 求解中的经典 RL 算法。

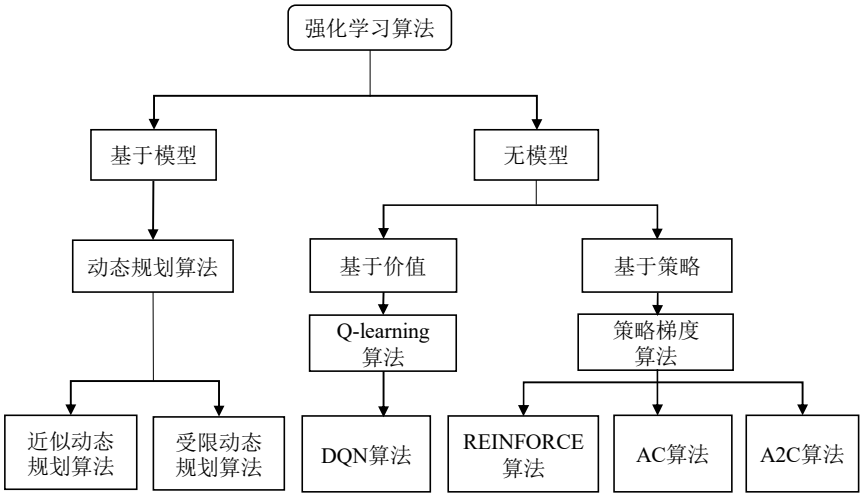

图 2-2 RL 算法分类图

RL 问题涉及智能体和环境之间的相互作用。在这个过程中，环境产生状态信息来描述系统状态，智能体通过观察这些状态并利用信息来做出决策。当智能体采取动作时，环境会相应地转到下一个状态，并返回奖励信号和下一个状态给智能体，以便智

能体可以通过这些信息来不断优化其决策策略。当（状态→动作→奖励）循环完成时，就说一个时间步已完成。循环重复，直到环境回路来描述。整个过程可以用图 2-3 来描述。

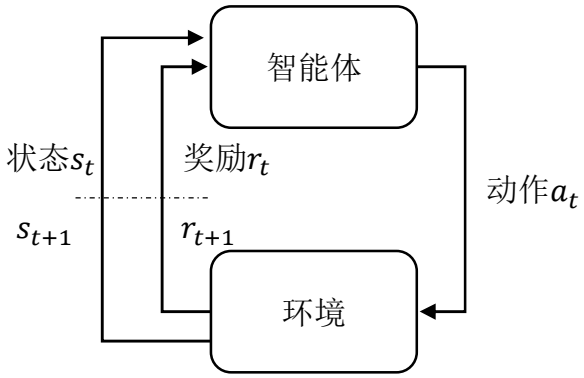

图 2-3 RL 原理

智能体的动作生成函数称为策略，RL 的目标是最大化智能体所获得的奖励之和，因此智能体需要选择最优的动作策略来实现这个目标，即智能体所获得的奖励之和，智能体通过选择好的动作使目标最大化。这需要智能体通过不断优化其策略来适应不同的环境，以便在长期的时间尺度上获得更高的收益。它通过在反复尝试和犯错的过程中与环境交互来学习如何做到这一点，并使用接收到的信号来强化好的动作。

本质上，RL 是一种基于反馈控制回路的学习方法，其核心思想是让智能体和环境相互作用并不断调整其动作，以实现最优决策并最大化奖励。根据奖励信号来调整动作以实现最优决策，在不断学习和适应中提高决策的准确性和效率，进而使目标最大化。交换的信号是（s_t, a_t, r_t），它们分别代表状态、动作和奖励，t 表示这些信号发生的时间步。（s_t, a_t, r_t）三元组称为经验。控制回路可以一直重复，直到到达终点状态或者最大时间步（t=T）时停止迭代。轨迹是一个事件的一系列经历，$\tau =$（s_0, a_0, r_0），（s_1, a_1, r_1）…通常情况下，智能体需要经历大量事件才能学习到一个有效的策略，所需事件数量因问题的复杂程度而异，范围从几百到几百万不等。

2.2.2 强化学习中的马尔可夫决策过程

考虑如何使用转换函数使环境从一个状态转换到下一个状态。在 RL 中，转换函数被表示为马尔可夫决策过程（Markov Decision Process, MDP），MDP 是一种常用的数学模型，用于对顺序决策进行建模。一般可以通过式（2-1）来理解为什么转换函数被表示为 MDP。

$$s_{t+1}\sim P\left(S_{t+1}\middle|(S_0,a_0),(S_1,a_1),\cdots,(S_t,a_t)\right) \quad (2\text{-}1)$$

式（2-1）表示在时间步 t，下一个状态s_{t+1}是从概率分布 P 中随机采样得到的，而这个概率分布 P 是基于整个历史的。如果一个事件持续了很多时间步，那么环境从一个状态s_t转移到另一个状态s_{t+1}的概率将取决于之前发生的所有状态 s 和动作 a。以这种方式对转换函数进行建模，需要考虑到多个状态和动作的影响，增加了建模的难度，是相当具有挑战性的，尤其是在事件跨越多个时间步的情况下。为了使环境的转化函数更加实用，将其转换为 MDP，下一个状态s_{t+1}只由前一个时刻的状态s_t和动作a_t决定，这就是马尔可夫特性。根据这种假设，新的转换函数可以表示为以下形式：

$$s_{t+1}\sim P\left(s_{t+1}\middle|(s_t,a_t)\right) \quad (2\text{-}2)$$

式（2-2）表示在下一个状态 s_{t+1} 是从概率分布 $P\left(s_{t+1}\middle|s_t,a_t\right)$中采样的，这是原始转换函数的一种简单形式，没有考虑更加复杂的因素和影响。马尔可夫性质表明，在某个时刻，系统的当前状态和已采取的动作所包含的信息足以确定下一个时刻状态的转移概率。

MDP 实际上就是一个多元组$[S,A,P,R,\gamma]$。其中，S 为所有状态空间的集合，$s_t\in S$ 表示智能体在 t 时刻所采取的动作；A 为智能体所执行动作的集合，$a_t\in A$表示智能体在 t 时刻所采取的动作，动作是对部分解的添加或者对完整解进行改变；$P(s_{t+1}|s_t,a_t)$ 为环境的状态转移概率矩阵；$R(s_t,a_t,s_{t+1})$是环境的奖励函数，表明为特定状态下选择的动作对解决方案造成的影响。$\gamma\in(0,1)$是折扣因子，调控智能体考虑短期回报。

2.3 深度强化学习原理概述

随着人类社会的迅速进步，越来越需要在各种复杂场景中使用深度学习来自动学习大规模输入数据的抽象表达形式，并以此为基础进行强化学习，从而优化问题的解决策略。由此，谷歌旗下的人工智能研究团队 DeepMind，将 DL 和 RL 巧妙得结合在一起，创造性地形成了深度强化学习（Deep Reinforcement Learning, DRL）这一全新的研究领域[58]，引起了广泛的关注和热议。DRL 属于 RL 中一个较大的领域。RL 的核心是函数逼近，在 DRL 中，函数是用深度神经网络学习的。RL 与有监督和无监督学习一起构成了机器学习的三种核心技术，每种技术在问题的表达方式和算法的数据学习

方式上都有所不同。DRL 是 DL 与 RL 的结合，由于其在处理高维、非线性问题方面的优越性能，越来越多的国内外学者正在探索将深度强化学习技术应用于车辆路径问题的解决方案中。

在 2.2.1 节中，了解到 RL 中可学习的三个主要函数。相应地，可以将 DRL 归结为以下三大类算法，分别是基于策略的方法（学习策略），基于价值的方法（学习价值函数）和基于模型的方法（学习模型）。此外，还存在一些组合方法，智能体可以通过同时学习多个函数来提高性能，例如同时学习值函数和模型。图 2-4 给出了每类方法中主要的 DRL 算法以及它们之间的关系。

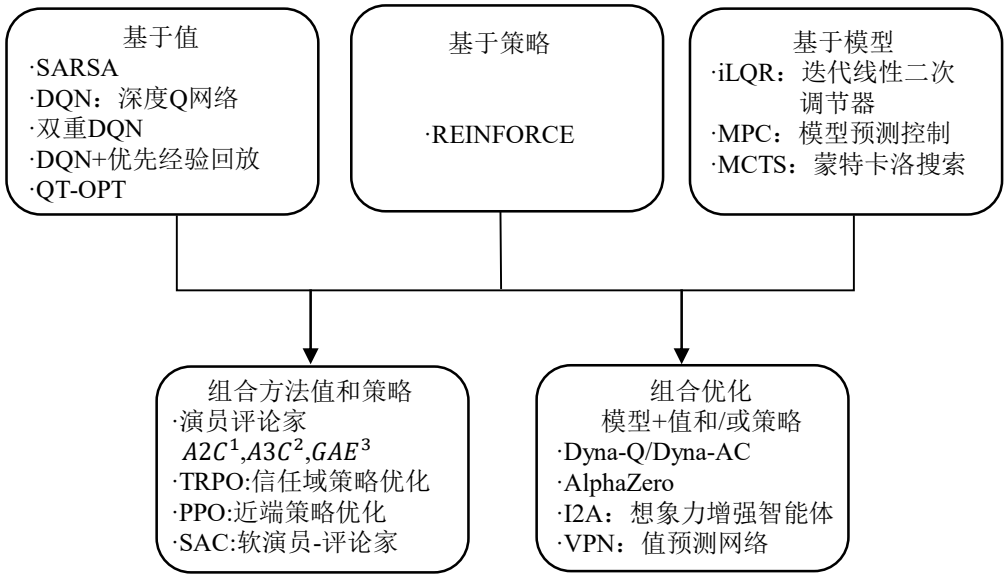

图 2-4 DRL 算法分类图

注：

1.A2C：优势演员-评论家

2.A3C：异步优势演员-评论家

3.GAE：带广义优势估计的演员-评论家

如图 2-5 所示，DRL 通过利用 DL 的非线性，对 RL 中的值函数、策略或者模型进行拟合，使得决策达到最好的效果，它的具体学习过程如下：

（1）智能体与环境相互作用获得高维度的观察信息，并采用 DL 算法对该信息进行感知和表达，以获得当前时刻状态的具体特征表示；

（2）根据预期回报对多种可行动作的价值函数进行全面评估，并据此评估结果采用相应的策略，将当前的状态转化为对应的动作表示；

（3）环境对当前的动作做出反应，以此获得下一个观察信息[58]。

通过上述过程的反复循环，通过不断得尝试和优化，智能体逐渐掌握如何在环境中获取高回报，最终可以得到实现目标的最优策略。

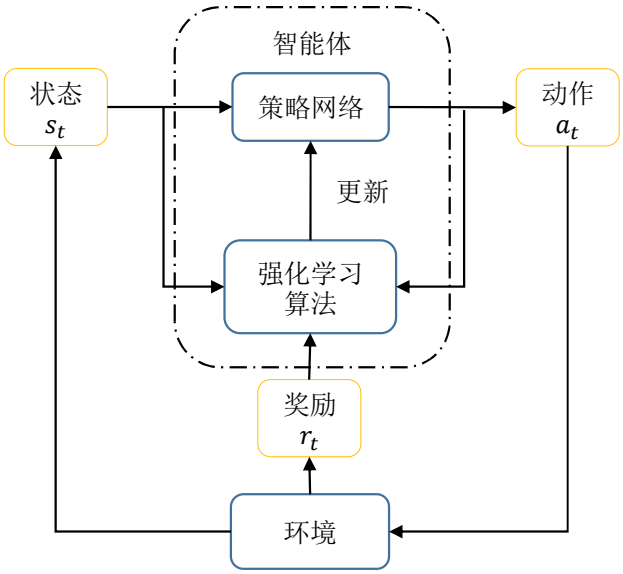

图 2-5 DRL 原理框架图

3 基于 DRL 求解异构车辆路径规划

在最简单的 VRP 中，一辆有装载有限的车辆负责将货物运送到多个客户节点，当装载完毕时，车辆必须返回到仓库来装载更多货物。目标是优化一组路线，以获得最大化的回报，所有路线的起点和终点都在仓库，该回报通常定义为车辆总成本的负值。即使是只有数百个客户节点，该问题的最优解也很难通过计算得到。相比同构 CVRP 中多个相同车辆的假设，不同容量（或速度）的车辆的设置更符合现实世界的实践，从而导致异构车辆路径问题[2,3]。根据目标，CVRP 也可以分别分为 min-max 和 min-sum。前者要求车队中车辆的最长（最坏情况）行驶时间（或距离）应尽可能令人满意，因为公平在许多实际应用中至关重要，而后者旨在最小化整个车队所发生的总行驶时间（或距离）[4]。具有 min-max 和 min-sum 目标的 HVRP 都是 NP-hard 问题，并且理论计算复杂程度随问题规模的扩大呈指数增长。

为了解决这个问题，提出了一种基于注意力机制的 DRL 方法，其中车辆选择解码器考虑了异构车队约束，节点选择解码器考虑了路线构建，该方法通过自动学习构建解决方案在每一步都选择车辆和该车辆的节点。所提出的方法可以有效地处理异构车队，并且优于大多数的启发式方法。

3.1 HVRP 数学模型

HVRP 问题描述了一个过程，即所有满载车辆从仓库出发，依次访问客户的位置以满足他们的需求，限制每个客户只能访问一次，以及单次行程的车辆装载量永远不能超过它的能力。本节主要介绍具有 min-max 和 min-sum 目标的 HVRP 的数学模型。

$N+1$ 个节点（客户和仓库）表示为 $\mathrm{X} = (x^i)_{i=0}^{N}$，节点$x^0$表示仓库，客户可以表示为$\mathrm{X}' = \mathrm{X} \setminus \{0\}$。每个客户节点$x^i \in R^3$定义为$\{(c^i, d^i)\}$，其中前者包含节点$x^i$的二维位置坐标，后者指其需求。具体来说，仓库$x^0 \in R^2$被定义为包含位置坐标的$\{c^0\}$。在这里，本文考虑了具有不同容量的异构车辆，这尊重了现实世界的情况。因此，设 $\mathrm{V} = \{v^i\}_{i=1}^{K}$ 表示车辆的异构车队，其中每个元素v^i定义为$\{Q^i\}$，即车辆v^i的容量。

数学模型中使用的变量定义如下：

- y_{ij}^k：如果车辆v^k直接从节点x^i行驶到节点x^j，则为 1，否则为 0；
- $\mathrm{D}(x^i, x^j)$：节点x^i和节点x^j之间的欧几里得距离；

- l_i^k：到本次访问为止，访问节点 i 的车辆v^k的路线上的累积需求；
- f：车辆的速度。

为简化起见，假设所有车辆都具有相同的速度f，可以很容易地将其扩展为采用不同的值。那么，MM-HVRP 目标公式可以定义如下：

$$\min \max_{v^k \in V} \left(\sum_{x^i \in X} \sum_{x^j \in X} \frac{D(x^i, x^j)}{f} y_{ij}^k \right) \tag{3-1}$$

受以下六个约束：

$$\sum_{x^i \in X} y_{ij}^k = \sum_{x^i \in X} y_{ji}^k, v^k \in V, x^j \in X \tag{3-2}$$

$$\sum_{v^k \in V} \sum_{x^i \in X} y_{ji}^k = 1, x^j \in X \tag{3-3}$$

$$l_j^k \geq l_i^k + d_j - Q^k(1 - y_{ji}^k), v^k \in V, x^i, x^j \in X' \tag{3-4}$$

$$d_j \leq l_i^k \leq Q^k, v^k \in V, x^i \in X' \tag{3-5}$$

$$y_{ij}^k \in \{0,1\}, y_{ii}^k = 0, v^k \in V, x^i, x^j \in X \tag{3-6}$$

$$d^i \geq 0, x^i \in X \tag{3-7}$$

目标是最小化所有车辆的最大行程时间。约束（3-2）确保车辆进入节点的次数等于它们离开节点的次数，其中可以多次访问仓库节点。约束(3-2)和约束(3-3)确保每个客户只被访问一次。约束（3-4）和约束（3-5）消除了次级行程，并保证任何车辆的累计需求不能超过其容量。约束（3-6）定义了二进制变量，约束（3-7）强加了变量的非负性。

MS-HVRP 与 MM-HVRP 共享相同的约束，而目标如下：

$$\min \sum_{v^k \in V} \sum_{x^i \in X} \sum_{x^j \in X} \frac{D(x^i, x^j)}{f} y_{ij}^k \tag{3-8}$$

其中f^k代表车辆v^k的速度，它可能因车辆不同而不同。因此，它实际上是在最小化整个异构车队的总行程时间。

3.2 基于注意力机制的 DRL 模型

在本节中，首先将 HVRP 的路径构建过程建模为 MDP，然后介绍基于 DRL 的方法，用于解决具有 min-max 和 min-sum 的 HVRP，最后描述训练策略网络的过程。策

略框架如图 3-1 所示。

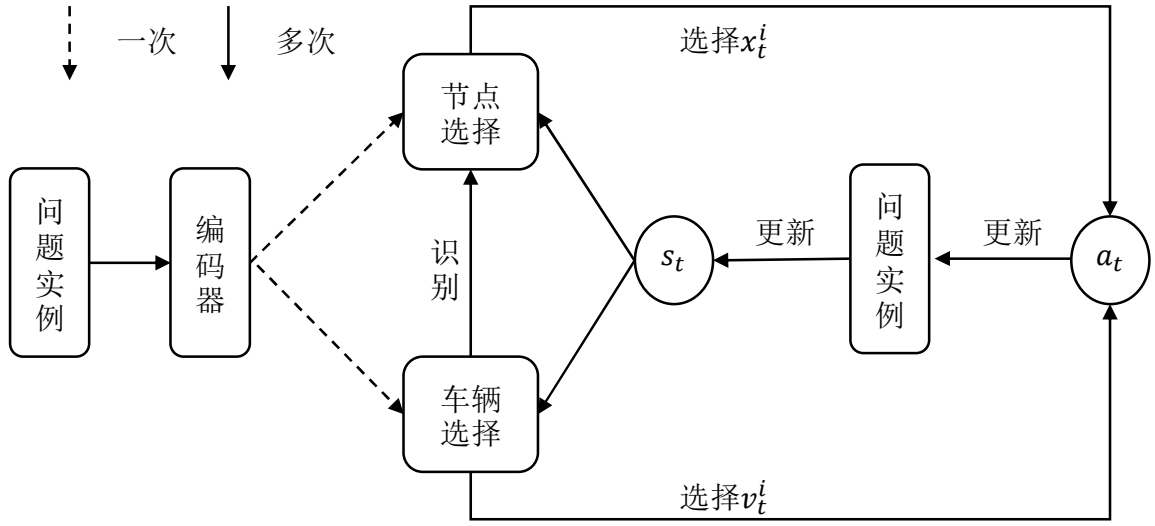

图 3-1 策略网络框架

3.2.1 马尔可夫决策过程模型

RL 最初是针对顺序决策问题提出的，例如自动驾驶汽车、机器人、游戏等[59]。HVRP 路径的逐步构建也可以看作是一个顺序决策问题。本节的工作中，将这样的过程建模为由 4 元组 M={S,A,τ,r}定义的马尔可夫决策过程[错误!未找到引用源。]。下面介绍状态空间 S、动作空间 A、状态转移规则τ、奖励函数 r 的详细定义。

状态：在 MDP 中，每个状态$s_t = (V_t, X_t)$由两部分组成。第一部分是车辆状态V_t，表示为$V_t = \{v_t^1, v_t^2, \cdots, v_t^K\} = \{(o_t^1, T_t^1, G_t^1), (o_t^2, T_t^2, G_t^2), \cdots, (o_t^K, T_t^K, G_t^K)\}$，其中$o_t^i$和$T_t^i$分别表示车辆$v^i$在步骤 t 的剩余装载能力和累计行驶时间。$G_t^i = \{g_0^i, g_1^i, \cdots, g_t^i\}$表示车辆$v^i$在步骤 t 的部分路线，其中g_t^i指的是车辆v^i在步骤 t 访问的节点。所有车辆的部分路线的维度（路线中的节点数）保持相同，即，如果在步骤 t 选择车辆v^i为节点x^j提供服务，其他车辆仍选择其最后服务的节点。离开仓库时（即$t = 0$），初始车辆状态设置为$V_0 = \{(Q^1, 0, \{x^0\}), (Q^2, 0, \{x^0\}), \cdots, (Q^K, 0, \{x^0\})\}$，其中$Q^i$是车辆$v^i$的最大容量。第二部分是节点状态$X_t$，表示为$X_t = \{x_t^0, x_t^1, \cdots, x_t^N\} = \{(c^0), (c^1, d_t^1), \cdots, (c^N, d_t^N)\}$，其中$c^j$是表示节点静态位置的二维向量，$d_t^i$是表示节点 i 在步骤 t $(d_t^0 = 0)$的需求的标量。此外，一旦为该节点提供服务，d_t^i将变为 0。这里不考虑需求拆分，只需要服务$d^i > 0$ 的节点。

动作：动作被定义为选择车辆和节点（客户或仓库）。具体来说，在$a_t \in A$处的动作表示为(v_t^i, x_t^i)即，所选节点x^j将在步骤 t 由车辆v^i服务（或访问），每一步仅选择一

辆车。此外，通过屏蔽方案屏蔽导致不可行解决方案的操作，以确保满足 HVRP 的所有约束。例如，如果节点的需求超过了满足容量约束的剩余负载能力，将屏蔽这些节点。

转移规则：转移规则τ将根据$a_t = (v_t^i, x_t^i)$时执行的动作，将前一个状态s_t转移到下一个状态s_{t+1}，即$s_{t+1} = (V_{t+1}, X_{t+1}) = \tau(V_t, X_t)$。车辆状态$V_{t+1}$中的元素更新如下：

$$o_{t+1}^k = \begin{cases} o_t^k - d_t^i, if\ k = i \\ o_t^k, otherwise \end{cases} \tag{3-9}$$

$$T_{t+1}^k = \begin{cases} T_t^k + \dfrac{D(g_t^k, x^j)}{f}, if\ k = i \\ T_t^k, otherwise \end{cases} \tag{3-10}$$

$$G_{t+1}^k = \begin{cases} [G_t^k, x^j], if\ k = i \\ [G_t^k, g_t^k], otherwise \end{cases} \tag{3-11}$$

其中g_t^k是G_{t+1}^k中的最后一个元素，即车辆v^k在步骤 *t* 最后访问的客户，$[\cdot,\cdot,\cdot]$是连接运算符。节点状态X_{t+1}中的元素更新如下：

$$d_{t+1}^i = \begin{cases} 0, if\ i = j \\ d_t^i, otherwise \end{cases} \tag{3-12}$$

其中每个需求在被访问后将保留为 0。

奖励：整个轨迹的奖励定义为目标成本的负值。假设目标函数被称为F_{MM}和F_{MS}，分别在方程（3-1）和方程（3-8）中为 MM-HVRP 和 MS-HVRP 定义。奖励由多个路径构建步骤累加，表示为$R_\tau \sum_{t=1}^{\tau} r_t$，其中$\tau$为路径构建步骤数，单步奖励$r_t$为当前动作引起的目标成本变化的负值。对于 MM-HVRP，$r_t = -\left(F_{MM}(s_t, a_t) - F_{MM}(s_t)\right)$，对于 MS-HVRP，$r_t = -\left(F_{MS}(s_t, a_t) - F_{MS}(s_t)\right)$。由于车辆可能需要多次返回仓库进行补给，因此$\tau$可能比 *N+1* 长。

3.2.2 策略网络框架

本文专注于学习由具有可训练参数θ的深度神经网络表示的随机策略$\pi_\theta(a_t|s_t)$。从初始状态s_0开始，即一个空解，遵循策略π_θ通过遵守第 2.2.2 节中的 MDP 来构造解，直到达到终止状态s_τ，即所有客户都由整个车队服务的车辆。因此，该过程基于链式法则表征如下：

$$\pi_\theta(s_\tau|s_0) = \prod_{t=0}^{\tau-1} \pi_\theta(a_t|s_t) \tag{3-13}$$

对于由编码器处理的实例的问题特征，策略网络首先使用车辆选择解码器选择车辆(v_t^i)，然后使用节点选择解码器为该车辆选择节点(x_t^j)，以便在每个路线构建步骤 t 访问，所选车辆和节点都构成了该步骤的动作，即$a_t = (v_t^i, x_t^j)$。其中状态相应更新。编码器执行一次，而车辆和节点选择解码器执行多次以构建解决方案。如图 3-1 所示，策略网络π_θ由编码器、车辆选择解码器和节点选择解码器组成。由于静态问题特定特征在整个决策过程中保持不变，因此编码器在第一步$(t = 0)$仅执行一次以简化计算，而其输出可以在后续步骤$(t > 0)$中重复用于路线构建。为了解决该实例，编码器处理问题特征以获得更好的表示，策略网络首先通过车辆选择解码器从整个车队中选择一辆车（v_t^i）并识别其索引，然后为此选择一个节点（x_t^i），在每个路线构建步骤中，车辆通过节点选择解码器访问。重复此过程，直到为所有客户提供服务。

3.2.3 策略网络架构

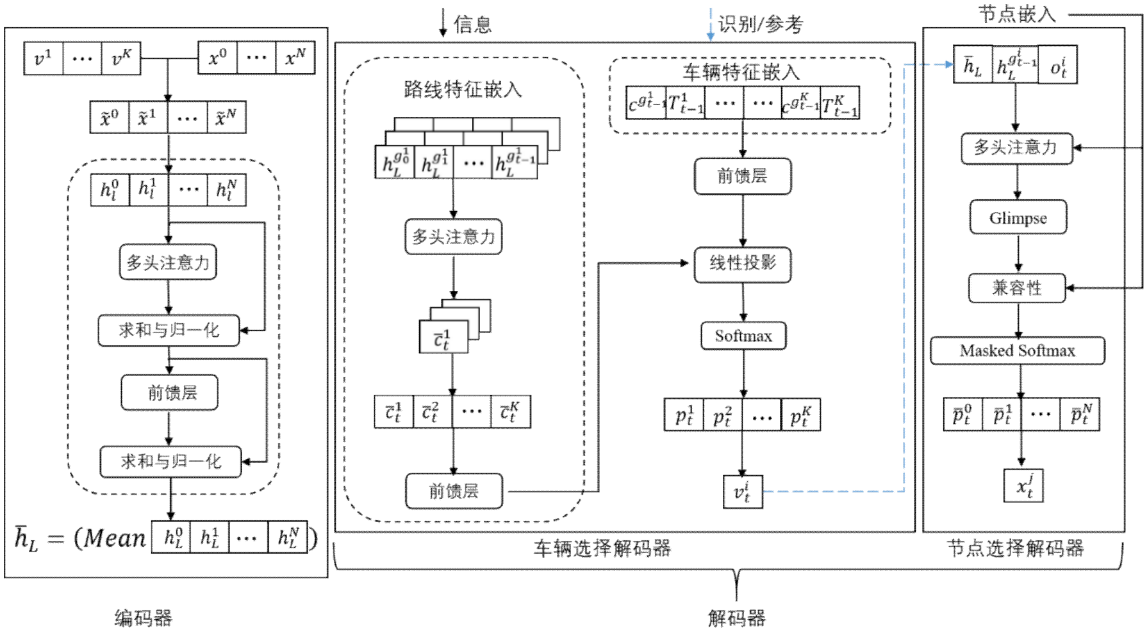

图 3-2 策略网络架构

如图 3-2 所示，策略网络（包含 K 个异构车辆和 N 个客户节点）采用编码器-解码器结构，解码器由两部分组成，即车辆选择解码器和节点选择解码器。基于任何车辆在每一步都有机会被选中的规定，策略网络能够在考虑到 HVRP 特性的情况下，在更合理、更广阔的行动空间中进行搜索。此外，通过添加从所有车辆和现有（部分）路线中提取的特征来丰富车辆选择解码器的上下文信息。它利用车辆特征（最后一个节

点位置和累积的旅行时间）、路线特征（m 辆车辆的最大路线池）及其组合来计算选择每辆车的概率。这样做时，车辆选择解码器允许策略网络捕获车辆的异构角色，以便从全局角度更有效地做出决策。接下来，分别介绍编码器、车辆选择解码器和节点选择解码器的细节。

编码器将问题特征（即客户位置、客户需求以及车辆容量）嵌入到更高维度的空间中，然后通过注意力层对其进行处理以更好地提取特征。通过除以每辆车的容量来规范化客户x^i的需求d_0^i，以反映车辆的差异，即$\tilde{x}^i = \left(c^i, d_0^i/Q^1, d_0^i/Q^2, \cdots, d_0^i/Q^K\right)$。增强的节点特征$\tilde{x}^i$然后线性投影到维度$d_h$=128 的高维空间中的$h_0^i$[48][48][48][48]。之后，通过 L 个注意力层进一步将h_0^i转换为h_L^i以获得更好的特征表示，每个注意力层由一个多头注意力子层和一个前馈子层组成，网络结构如图 3-3 所示。

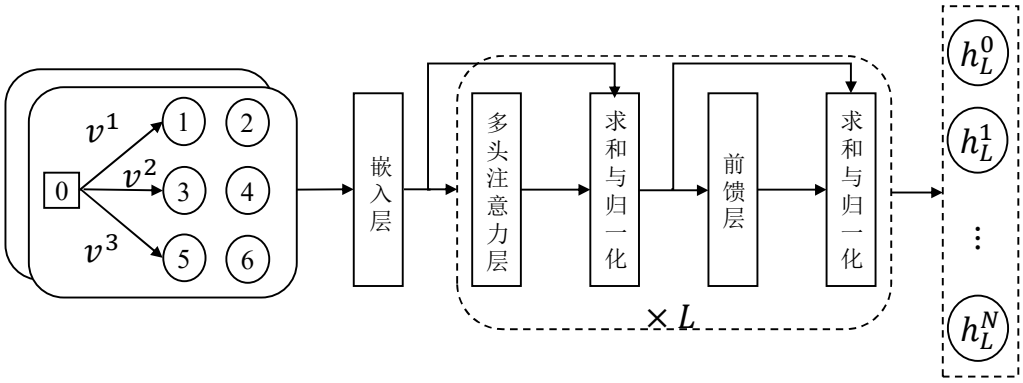

图 3-3 编码器网络结构

第 l 个 MHA 子层使用多头自注意力网络来处理节点嵌入$h_l = \left(h_l^0, h_l^1, \cdots, h_l^N\right)$。规定$d_q = d_k = d_h/M$是 query 和 key 维度，$d_v = d_h/M$是 value 维度，$M = 8$ 是注意力中的头数量。第 l 个 MHA 子层首先计算每个头$m \in \{1,2,\cdots,M\}$的注意力值$Z_{l,m}$，然后将所有这些头连接起来，并将它们投影到与输入h_l。具体来说，将这些步骤展示如下：

$$Q_{l,m} = h_l W_{l,m}^Q, K_{l,m} = h_l W_{l,m}^K, V_{l,m} = h_l W_{l,m}^V \tag{3-14}$$

$$h_{l,m} = Softmax\left(\frac{\left(Q_{l,m}\right)^{\mathrm{T}}\left(K_{l,m}\right)}{\sqrt{d_k}}\right)V_{l,m} \tag{3-15}$$

$$\begin{aligned}\mathrm{MHA}(h_l) &= \mathrm{MHA}\left(h_l W_l^Q, h_l W_l^K, h_l W_l^V\right) \\ &= \left[h_{l,1}, h_{l,2}, \cdots, h_{l,M}\right]W_l^O\end{aligned} \tag{3-16}$$

其中$W_l^Q \in R^{M\times d_h\times d_q}$、$W_l^k \in R^{M\times d_h\times d_k}$、$W_l^V \in R^{M\times d_h\times d_v}$和$W_l^Q \in R^{M\times d_h\times d_h}$是第 l 层中的可训练参数，并且在不同的注意力层之间是独立的。

之后，第 l 个 MHA 子层的输出被馈送到具有 ReLU 激活函数的第 l 个前馈子层，以获得下一个嵌入h_{l+1}。这里，跳过连接和批量归一化（Batch Normalization, BN）层用于 MHA 和前馈子层，总结如下：

$$h_l^{'} = BN\left(h_l + MHA(h_l)\right) \tag{3-17}$$

$$\mathrm{h}_{l+1} = BN\left(h_l^{'} + FF\left(h_l^{'}\right)\right) \tag{3-18}$$

最后，定义编码器的最终输出，即h_L^i，$x^j \in X$，作为节点嵌入；节点嵌入的平均值，即，$\bar{h}_L = \frac{1}{N+1}\sum_{i=0}^{N} h_L^i$，作为图嵌入问题实例，它将在解码器中重复使用多次。解码器的网络结构如图 3-4 所示：

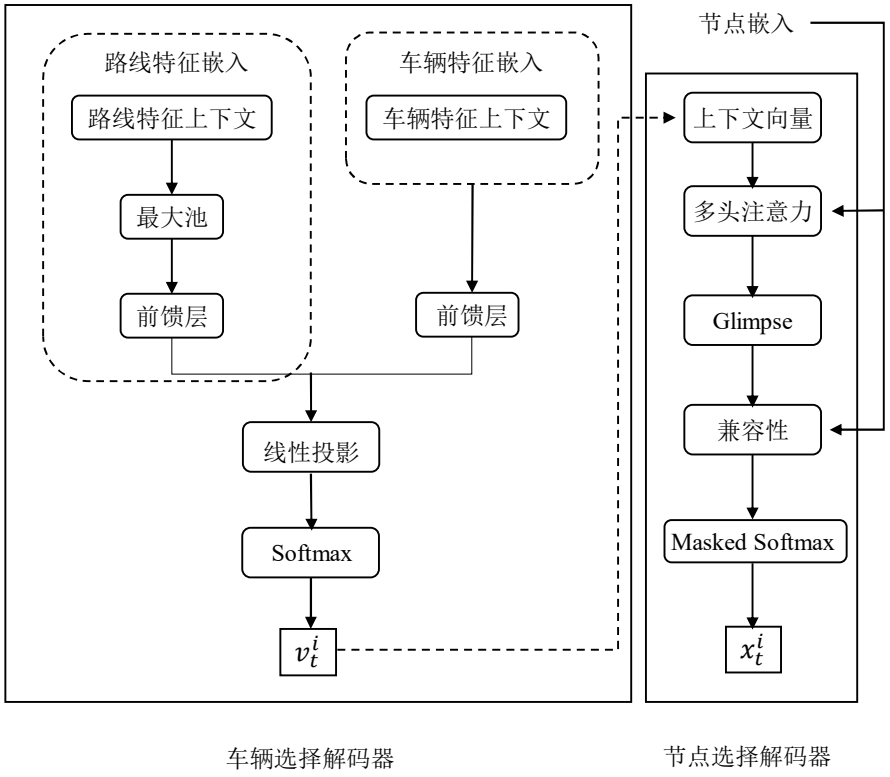

图 3-4 解码器网络结构

车辆选择解码器：车辆选择解码器输出选择特定车辆的概率分布，主要利用两种嵌入，即车辆特征嵌入和路线特征嵌入。

1）车辆特征嵌入：为了捕获当前步骤中每辆车的状态，在步骤 t 定义车辆特征上下文$C_t^V \in R^{1\times 3K}$如下所示：

$$C_t^V = \left[C^{g_{t-1}^1}, T_{t-1}^1, C^{g_{t-1}^2}, T_{t-1}^2, \cdots, C^{g_{t-1}^K}, T_{t-1}^K\right] \tag{3-19}$$

其中$c^{g_{t-1}^i}$表示车辆v^i在步骤 *t*-1 的部分路线中最后一个节点g_{t-1}^i的二维位置，T_{t-1}^i是车辆v^i到步骤 *t*-1 的累积行驶时间。然后，使用可训练参数 W_1 和 b_1 线性投影车辆特征上下文，并由 512 维前馈层进一步处理，该层具有 ReLU 激活功能，以在步骤 *t* 生成车辆特征嵌入H_t^V，如下所示：

$$H_t^V = FF\left(W_1 C_t^V + b_1\right) \tag{3-20}$$

2）路线特征嵌入：路线特征嵌入从所有车辆的现有部分路线中提取信息，这有助于策略网络从先前步骤中的访问节点中进行本质上的学习，而不是像以前的研究中那样简单地屏蔽它们[40]。对于步骤 *t* 中的每个车辆v^i，将其路线特征上下文$\widetilde{C}_t^i$定义为节点嵌入的排列（即h_L^j是节点x^j的节点嵌入），对应于其部分路线G_{t-1}^i中的节点。具体来说，每个车辆v^i，$i = 1,2,\cdots,K$的路线特征上下文$\widetilde{C}_t^i$定义如下所示：

$$\widetilde{C}_t^i = \left[h_L^{g_0^i}, h_L^{g_1^i}, \cdots, h_L^{g_{t-1}^i}\right] \tag{3-21}$$

其中$\widetilde{C}_t^i = R^{t \times d_h}$（第一个维度的大小为 *t*，因为$G_{t-1}^i$在步骤 *t* 应该有 *t* 个元素），$h_L^{g_{t-1}^i}$表示车辆$v^i$部分路径$G_{t-1}^i$中第$t-1$个节点的$h_L$中的相应节点嵌入。例如，假设$t = 4$，车辆$v^i$的部分路线是$G_{t-1}^i$，那么这辆车在步骤$t = 4$的路线特征上下文将是$\widetilde{C}_4^i = \left(h_L^{g_0^i}, h_L^{g_1^i}, h_L^{g_2^i}, h_L^{g_3^i}\right) = \left(h_L^0, h_L^3, h_L^3, h_L^1\right)$。

之后，所有车辆的路线特征上下文通过最大池聚合，然后连接以产生整个车队的路线上下文$\widehat{C}_t^R$，然后通过具有可训练参数W_2和b_2以及 512 维前馈层的线性投影进一步处理，以生成在步骤t的路线特征嵌入H_t^R，如下所示：

$$\overline{C}_t^i = max\left(\widetilde{C}_t^i\right), i = 1,2,\cdots,K \tag{3-22}$$

$$\widehat{C}_t^R = [\overline{C}_t^1, \overline{C}_t^2, \cdots, \overline{C}_t^K] \tag{3-23}$$

$$H_t^R = FF\left(W_2 \widehat{C}_t^R + b_2\right) \tag{3-24}$$

最后，车辆特征嵌入H_t^V和路线特征嵌入H_t^R与参数W_2和b_2连接并线性投影，通过 *Softmax* 函数进一步处理以计算概率向量，如下所示：

$$H_t = W_3\left[H_t^V, H_t^R\right] = b_3 \quad (3\text{-}25)$$

$$p_t = Softmax(H_t) \quad (3\text{-}26)$$

其中$p_t \in R^K$，其元素p_t^i表示在时间步t选择车辆v^i的概率。根据不同的策略，可以通过贪婪地检索最大概率来选择车辆，也可以根据向量 P_t 进行采样。然后将所选车辆v^i用作节点选择解码器的输入。

节点选择解码器：给定来自编码器的节点嵌入和来自车辆选择解码器的所选车辆v^i，节点选择解码器输出所有未访问节点（前面步骤中服务的节点被屏蔽）的概率分布$\bar{p}_t$，用于识别所选车辆要访问的节点。首先定义一个上下文向量H_t^c，它由图嵌入$\bar{h}_L$、所选车辆访问的最后（前一个）节点的节点嵌入和该车辆的剩余负载能力组成[44]，如下所示：

$$H_t^c = \left[\bar{h}_L, h_L^{g_{t-1}^i}, o_t^i\right] \quad (3\text{-}27)$$

其中第二个元素$h_L^{g_{t-1}^i}$,与等式（3-21）中定义的含义相同，指的是$t = 0$ 时仓库的节点嵌入。设计的上下文向量突出了当前决策步骤中所选车辆的特征，并从全局角度考虑了实例的图嵌入。然后将上下文向量H_t^c和节点嵌入h_L馈入多头注意力层以合成新的上下文向量$\widehat{H}_t^c$，作为节点嵌入的 glimpse[61]。与编码器中的自注意力不同，这个注意力的 query（查询）来自上下文向量，而注意力的 key/value 来自节点嵌入，如下所示：

$$\widehat{H}_t^c = MHA\left(H_t^c W_c^Q, h_L W_c^K, h_L W_c^V\right) \quad (3\text{-}28)$$

其中W_c^Q、W_c^K和W_c^V是类似于等式（3-16）的可训练参数。然后通过比较增强上下文$\widehat{H}_t^c$和通过兼容层嵌入的节点h_L之间的关系来生成概率分布$\bar{p}_t$。步骤t中所有节点与上下文的兼容性计算如下：

$$u_t = C \cdot \tanh\left(\frac{{q_t}^{\mathrm{T}} k_t}{\sqrt{d_k}}\right) \quad (3\text{-}29)$$

其中$q_t = \widehat{H}_t^c W_{comp}^Q$和$k_t = h_L W_{comp}^K$是可训练的参数，$C$ 设置为 10 来控制u_t的熵。最后，概率向量在方程式（3-30）中计算。其中，前面步骤中访问的所有节点都被屏蔽，以确保可行性,元素$\bar{p}_t^j$表示在步骤t选择由所选车辆v^i服务的节点x^i的概率，如下所示：

$$\bar{p}_t = Softmax(u_x) \quad (3\text{-}30)$$

与车辆选择的解码策略类似，节点可以通过始终检索最大$\bar{p}_t^j$来选择，或者根据向量$\bar{p}$以不那么贪婪的方式进行采样。

为了更好地说明所提出的方法，图 3-3 和图 3-4 分别展示了编码器和解码器结构在两个实例上的示例，其中包含七个节点和三个车辆，在此示例中，节点和车辆的特征通过编码器进行处理，以计算节点嵌入和图嵌入。在车辆选择解码器中，当前状态 s_t 下的三辆车的三个行程的节点嵌入，即$\left\{\{h^0,h^1,h^1,h^1\},\{h^0,h^0,h^3,h^3\},\left\{h^0,h^0,h^0,h^5\right\}\right\}$（假设前三步轮流选择三辆车），进行路线特征提取处理，处理三辆车的当前位置和累计行驶时间进行车辆特征提取，然后将它们拼接起来计算选择车辆的概率。在本例中，对于选定的车辆v^1，首先将添加了图嵌入的当前节点嵌入h^1和该车辆的当前加载能力进行连接并线性传播，进一步用于计算选择具有掩码 *Softmax* 的节点的概率，即，$\bar{p}^1=\bar{p}^3=\bar{p}^5$。在本例中选择节点$x^2$，动作表示为$a_t=\{v^1,v^2\}$，状态更新并转换为$s_{t+1}$。

3.2.4 策略梯度

模型的具体训练过程如算法 1。

算法 1 REINFORCE 算法

输入：策略网络π_θ的初始参数θ；baseline 网络$\pi_{\theta^{BL}}$的初始参数$\theta^{BL}=\theta$；epoch 的数量 E；batches 的矢量 I；batch 大小 B；最大路线构建步骤Γ。

for epoch = 1,2,⋯,*E* **do**
  for i =1,2,⋯,*I* **do**
    批量随机生成 B 个训练实例
    for t = 0,1,⋯,Γ **do**
      选择一个动作$a_{t,b}\sim\pi_{\theta(a_{t,b}|s_{t,b})}$,b ∈ {1,2,⋯,$B$}
      获得奖励$r_{t,b}$和下一个状态s_{t+1},b ∈ {1,2,⋯,B}
    end for
    $R_b=\sum_{t=0}^{\Gamma}r_{t,b}$,b ∈ {1,2,⋯,$B$}
    带有基线的 GreddyRollout $\pi_{\theta^{BL}}$获得基线奖励$R_{b^{BL}}$,b ∈ {1,2,⋯,B}
    $d_\theta\leftarrow\frac{1}{B}\sum_{b=1}^{B}\left(R_b-R_b^{BL}\right)+\nabla_\theta log\pi_\theta\left(s_{\Gamma,b}|s_{0,b}\right)$,b ∈ {1,2,⋯,$B$}
    θ ← Adam(θ,d_θ)
  end for
  if $\text{O}_{\text{NE}}\text{S}_{\text{IDED}}\text{P}_{\text{AIRED}}\text{TT}_{\text{EST}}\left(\pi_\theta,\pi_{\theta^{bl}}<\alpha\right)$ **then**
    $\theta^{BL}\leftarrow\theta$
  end if
end for

算法 1 列出了所提出的 DRL 方法，采用带 baseline 的策略梯度来训练车辆选择和节点选择策略以进行路线构建。策略梯度由两个网络表征：（1）策略网络，即前面提到的策略网络π_θ，在每个解码步骤中选择一个动作并为车辆和节点生成关于该动作的概率向量；（2）baseline 网络$\pi_{\theta^{BL}}$，一个贪婪的 rollout 基线，其结构与策略网络相似，但始终通过选择具有最大概率的车辆和节点来计算奖励。用蒙特卡罗方法来更新参数以迭代地改进策略。

其中 GreedyRollout 表示对该模型使用取最大选择概率的节点的策略得到解的神经

网络。在每个 epoch，为每个问题实例构建路线并在第 8 行计算关于该解决方案的奖励，并在第 11 行更新策略网络的参数。此外，基线网络R_b^{BL}的预期奖励来自第 9 行中策略的贪婪 rollout。根据第 14 行中多个实例进行配对 t-test，如果最新策略网络的性能显著优于前者，则基线网络的参数将被最新的策略网络的参数所取代。通过更新这两个网络，策略π_θ不断进行迭代改进，旨在找到更高质量的解决方案。

3.3 实验结果与分析

在本章中，进行实验来评估 DRL 方法。特别是，由不同容量的满载车辆组成的异构车队，从一个仓库节点出发，按照一定的路线出发以满足所有客户的需求，其目标是最大限度地减少车辆的最长或总行程时间。

3.3.1 实验设置

为实验描述了设置和数据生成方法，对于 MM-HVRP，仓库和客户的坐标使用均匀分布在单位正方形$[0,1] \times [0,1]$内随机抽样。客户的需求是从集合$\{1,2,\cdots,9\}$中随机选择的离散数（仓库需求为 0）[40,48]。为了全面验证性能，考虑了异构车队的设置。车队考虑三辆异构车辆（名为 V_3），其容量分别设置为 20、25 和 30。本文方法针对车队的不同客户规模进行了评估，其中考虑 V_3 的 40、60、80、100；在 MM-HVRP 中，为了简化，将所有车辆的车速 f 设置为 1.0。然而，DRL 方法能够应对不同的速度，这在 MS-HVRP 中得到了验证。MS-HVRP 的大部分设置与 MM-HVRP 相同，只是车辆速度与其容量成反比。这样做可以避免只选择容量最大的车辆来服务所有客户，从而最大限度地减少总行程时间。将 V_3 的速度分别设置为 1/4、1/5 和 1/6。

共享超参数以针对所有问题大小训练策略。训练实例是动态随机生成的，epoch 大小为 1,280,000，每个 epoch 分为 2500 个批次[62]。关于 epoch 的数量，通常更多的 epoch 会带来更好的性能。但是，在经过大量 epoch 的训练之后，如果性能的提升不是很大，可以在完全收敛之前停止训练，这仍然可以提供有竞争力的性能，尽管不是最好的。本章实验中，对所有问题大小使用 50 个 epoch 来证明本文方法的有效性，而在实践中可以采用更多的 epoch 以获得更好的性能。节点和车辆的特征被嵌入到 128 维空间中，然后再输入车辆选择和节点选择解码器[62]。此外，使用 Adam 优化器来训练策略参数，初始学习率为 10^{-4}，每个 epoch 衰减 0.995 以实现收敛。所有梯度向量的范数都被限制在 3.0 以内，第 3.2.3 节中的α 设置为 0.05。关于测试，从均匀分布中为每个问题大小

随机生成 1,280 个实例，并且对于 DRL 方法和基线是固定的。所有实验均在相同的实验环境下的同一机器上运行，机器配置为 Inter（R） Core（TM） i7-11700 CPU，NVIDIA GeForce RTX 3060 Ti GRU，Windows10 操作系统，python 语言实现。

3.3.2 比较分析

对于 MM-HVRP，寻找最佳解决方案非常耗时，尤其是对于大型问题。因此，采用各种改进的经典启发式方法作为基线，其中包括：（1）通过删除字符串进行松弛归纳（Slack Induction by String Removals, SISR）[63]，这是一种针对 CVRP 及其变体的强启发式方法，在目标值和差距方面优于启发式算法优化器（Lin Kernighan Helsgaun 3, LKH3）；（2）可变邻域搜索（Variable Neighborhood Search, VNS），一种解决一致 VRP 的有效启发式方法[64]；（3）蚁群优化（Ant Colony Optimization, ACO），蚁群系统的改进版本，用于求解具有时间窗的 HVRP[65]，并行运行所有蚂蚁的解决方案构建以减少计算时间；（4）萤火虫算法（Firefly Algorithm, FA），标准 FA 方法的改进版本，用于解决异构固定车队车辆路径问题[66]。本文调整了所有基线的目标和相关设置，以便它们与 MM-HVRP 共享相同的基线。

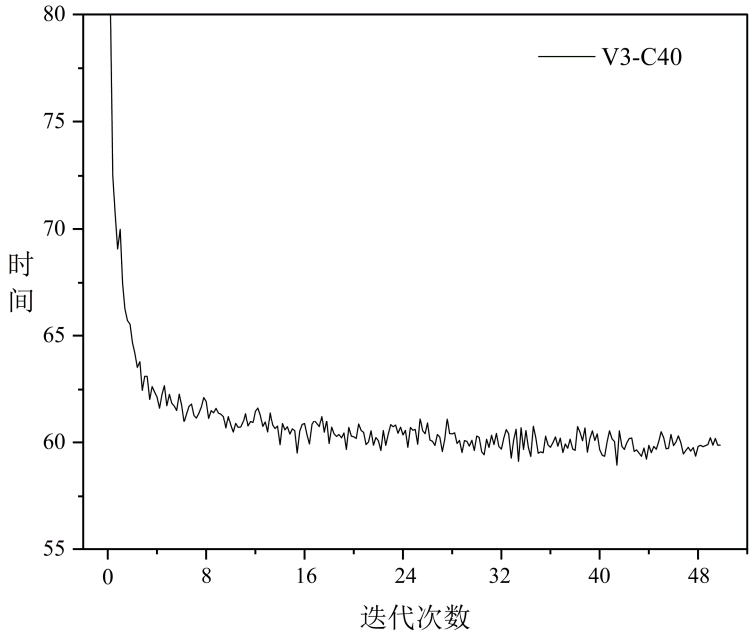

图 3-5 V3-C40 min-sum 目标曲线

关于迭代，线性增加 VNS、ACO 和 FA 的原始迭代，因为问题规模扩大以获得更好的性能，而原始设置在所有问题规模上采用相同的迭代。对于 SISR，遵循其原始设置，随着问题规模的增大，迭代次数也随之增加。对于 MS-HVRP，具有与 MM-HVRP

相同的四个启发式基线[67]。

在与其他方法进行比较之前，首先评估 DRL 的收敛性，这可以通过图 3-5 来说明，即 V3-C40，其中，V3-C40 指的是 3 辆汽车和 40 个客户（包括仓库在内的 41 个节点），该解释适用于所有此类表达。从曲线中看到目标值下降非常快，直到收敛，这表明 DRL 方法在最小化车队总成本的收敛策略方面表现良好。本文采用了两种不同的动作选择策略：

（1）在贪婪动作选择策略中，基于对策略网络的完全信赖，每一步操作都以解码器输出概率值为基准，选择具有最大概率的动作（车辆和节点），称为 Ours（Greedy）；

（2）采样动作选择策略基于解码器输出概率分布进行动作采样选择，而非仅仅选择概率值最大的动作。因此，该策略在每次选择动作时，会以不同的概率选择各个可能的动作。根据式（3-26）和式（3-30）中计算的概率进行抽样，产生 S 解，然后检索最好的。将 S 设置为 1,280 和 12,800，并将它们分别称为 Ours（S=1,280）和 Ours（S=12,800）。

表 3-1 DRL 方法和基线对比

	方法	V3-C40		V3-C60		V3-C80		V3-C100		V3-C120	
		Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap	Obj.	Gap
Min-max	SISR[63]	4.00	0%	5.58	0%	7.27	0%	8.89	0%	10.42	0%
	VNS[64]	4.17	4.25%	5.80	3.94%	7.57	4.13%	9.20	3.49%	10.81	3.74%
	ACO[65]	4.31	7.75%	6.18	10.75%	8.14	11.97%	10.05	13.05%	11.79	13.15%
	FA[66]	4.49	12.25%	6.30	12.90%	8.32	14.44%	10.11	13.72%	11.98	14.97%
	OURS(Greedy)	4.44	11.00%	6.09	9.13%	7.79	7.15%	9.46	6.40%	10.90	4.60%
	OURS(S=1280)	4.17	4.25%	5.72	2.50%	7.50	3.16%	9.06	1.9%	10.59	1.63%
	OURS(S=12800)	4.14	3.75%	5.69	1.90%	7.46	2.61%	9.00	1.23%	10.50	0.77%
Min-sum	SISR[63]	55.79	0%	79.12	0%	103.41	0%	126.19	0%	149.10	0%
	VNS[64]	57.54	3.13%	81.44	2.93%	106.18	2.68%	129.32	2.48%	152.56	2.32%
	ACO[65]	60.11	7.74%	86.05	8.76%	113.75	11.06%	140.61	12.84%	166.50	11.67%
	FA[66]	59.94	7.43%	85.36	7.89%	112.81	10.00%	138.92	11.36%	164.53	10.35%
	OURS(Greedy)	58.93	5.63%	83.03	4.94%	108.44	4.86%	131.69	4.36%	154.52	3.64%
	OURS(S=1280)	57.08	2.31%	80.46	1.69%	105.26	1.79%	128.59	1.90%	151.25	1.44%
	OURS(S=12800)	56.86	1.92%	79.89	0.97%	104.65	1.20%	128.23	1.62%	150.76	1.11%

表 3-2 DRL 方法和基线的计算时间对比

	方法	V3-C40（s）	V3-C60（s）	V3-C80（s）	V3-C100（s）	V3-C120（s）
Min-max	SISR[63]	245	468	752	1135	1657
	VNS[64]	115	294	612	927	1378
	ACO[65]	209	317	601	878	1242
	FA[66]	168	285	397	522	667
	OURS(Greedy)	0.70	0.82	1.11	1.44	1.94
	OURS(S=1280)	1.25	1.43	2.25	3.42	4.52
	OURS(S=12800)	1.64	2.97	4.56	6.65	8.78
Min-sum	SISR[63]	254	478	763	1140	1667
	VNS[64]	109	291	547	828	1217
	ACO[65]	196	302	593	859	1189
	FA[66]	164	272	388	518	653
	OURS(Greedy)	0.61	1.02	1.11	1.56	1.96
	OURS(S=1280)	1.18	1.49	2.34	3.38	4.61
	OURS(S=12800)	1.65	2.99	4.63	6.74	9.11

表 3-1 中记录了 DRL 方法和基线在所有规模的 MM-HVRP 和 MS-HVRP 实例上的性能，其中包括平均目标值（Obj.）和最优差距（Gap）。每个实例相对于这些方法的计算时间分别记录在表 3-2。鉴于优化求解 MM-HVRP 非常耗时，这里的差距是通过将一种方法的目标值与所有方法中找到的最佳值进行比较来计算的。

结合这两个表，可以看出，虽然采样 S=1,280 和 S=12,800 都比贪婪策略计算时间稍长，但实现了更小的目标值和差距，这证明了采样策略在提高解决方案质量方面的有效性。Ours（Greedy）在目标值和差距方面优于 FA，Ours（S=12,80）在目标值和最优差距方面可以超过所有的 ACO。在大多数情况下，Ours（S=12,80）优于 VNS，但在 V3-C40 情况下除外，此时两者最优差距相同。Ours（S=12,800）在 MM-HVRP 和 MS-HVRP 上的整体性能均优于 VNS、ACO、FA，并且与 SISR 相比也表现出色。可以得出结论，DRL 方法在解决 HVRP 方面比传统方法更有效，而且优于大多数比较启发式方法，并且与基线启发式方法 SISR 相比，具有令人满意的计算时间。

表 3-3 DRL 方法对比 CVRPLIB 基线

	分布	实例	最优目标	DRL	VNS[64]	SISR[63]
Min-max	均匀	P-n60-k10	-	306	308	293
		A-n61-k9	-	318	307	299
		E-n76-k7	-	372	375	362
		A-n80-k10	-	793	813	776
		E-n101-k8	-	448	455	428
		Avg.Gap	-	4.13%	4.49%	0%
	非均匀	B-n41-k6	-	384	371	359
		B-n51-k7	-	396	378	369
		B-n63-k10	-	565	558	540
		M-n101-k10	-	419	401	391
		CMT11	-	878	869	858
		Avg.Gap	-	5.46%	2.59%	0%
Min-sum	均匀	P-n60-k10	4009*	4054	4265	4.13
		A-n61-k9	3984*	4039	4252	3995
		E-n76-k7	4740*	5035	5222	4847
		A-n80-k10	11149*	11456	11466	11186
		E-n101-k8	5653*	5972	6114	5727
		Avg.Gap	0%	2.08%	5.51%	1.28%
	非均匀	B-n41-k6	4948*	5327	5015	4948
		B-n51-k7	5235*	5434	5363	5236
		B-n63-k10	7706*	7806	7825	7727
		M-n101-k10	5443*	5704	5687	5507
		CMT11	-	12528	12183	11910
		Avg.Gap	0%	4.55%	2.42%	0.29%

为了全面评估 DRL 方法的性能，进一步对应用训练的模型来解决从 CVRPLIB 基准中随机选择的实例，该基准是文献中用于算法比较的 VRP 实例的著名在线基准库。选择 10 个实例，并通过采用实例的客户位置和需求，使其适应本文 MM-HVRP 和 MS-HVRP 设置，其中一半遵循关于客户位置的均匀分布，其余一半不遵循。在表 3-3 中，记录了 CVRPLIB 上的比较结果，对于 DRL 方法，直接利用表 3-1 和 3-2 中的训

练模型来求解 CVRPLIB 实例，其中采用与实例最接近的模型。例如，使用针对 V3-C60 训练的模型求解 B-n63-k10。

本文选择 SISR 和 VNS 作为 MM-HVRP 和 MS-HVRP 的基线，它们在之前的实验中比其它比较启发式方法表现更好。每个目标值都是用不同的随机种子进行 10 次独立运行的平均值。从表 3-3 中可以观察到，DRL 方法在均匀分布的实力上最优性差距往往优于 VNS，而在非均匀分布的实例上表现稍差。此外，DRL 方法与高度优化的 SISR 方法相比也具有竞争力。如果参考 DRL 方法与 MS-HVRP 的精确方法和 MM-HVRP 的 SISR 方法之间的差距，本文的方法在均匀分布的实例上往往比非均匀分布的实例表现更好。

这种关于不同分布的观察是有实际意义的，特别是考虑到如第 3.3.1 节所述，所有的训练实例中的客户位置都遵循均匀分布，该设置在这一项研究领域被广泛采用。由于本文的 DRL 模型本质上是一种学习方法，因此当训练和测试实例都来自相同（或相似）均匀分布时，它确实具有提供卓越性能的潜力。

3.4 本章小结

本章提出了一种神经构造启发式方法，该方法结合了一个负责异构车队的车辆选择解码器和一个负责路线构建的节点选择解码器。通过这样做，所提出的方法可以通过异构车队自动灵活地选择车辆，然后在每个步骤中为该车辆访问一个节点。实验结果表明，DRL 方法可以有效地处理异构车辆并实现更低的最优性差距，计算时间稍长。此外，所提出的方法优于大多数比较启发式方法，并且与 SISR 方法相比，具有令人满意的计算时间。此外，扩展实验结果表明，该方法也能很好地求解 CVRPLIB 实例，性能令人满意。

4 基于 DRL 求解有容量限制的电动车辆路径规划

随着人们环保意识的加强，以降低运输过程对环境的影响作为优化目标的车辆路径问题逐渐受到各国研究人员的重视。电动汽车（Electric Vehicles, EV）是目前比较环保的交通工具，它不是由化石燃料驱动的内燃机，从而有效减少了交通运输过程中的碳排放，缓解了环境压力，在城市交通和物流系统中发挥着越来越重要的作用。前一章提出了一种基于深度强化学习的方法解决异构有容量限制的车辆路径问题，在此背景下，电动汽车路径问题考虑了车辆的具体特性，例如车辆的自主性和电池充电需求等。为此，在前章所提出的 DRL 框架的基础上，加入图嵌入组件及全局信息，实现图的局部信息和整体信息综合定义，用于求解有容量限制的电动车辆路径问题（Capacitated Electric Vehicle Routing Problem, CEVRP）。所提出的模型能够快速捕获图中嵌入的重要信息，进而有效地为问题提供相对较好的可行解。

4.1 CEVRP 数学模型

CEVRP 是车辆路径问题的变体，目的是在安排车辆路径时考虑减少 CO_2 和 NOx 等温室气体排放，减少对人类健康的负面影响。在 CEVRP 目标函数中考虑任何类型的排放，主要关注最小化路线成本和污染排放。这些污染物的数量与车辆消耗的燃料量成正比，因此，减少燃料消耗将有助于减少污染。

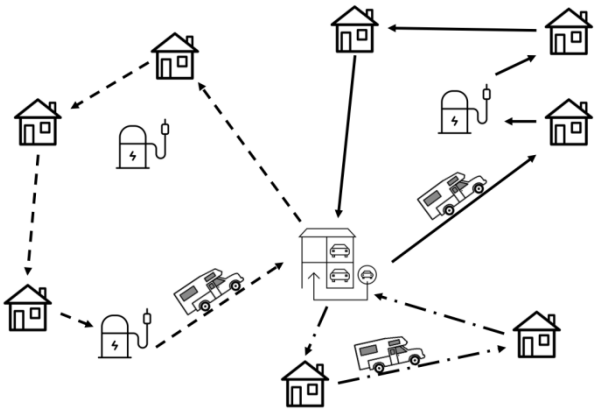

图 4-1 电动车辆路径问题

CEVRP 是一个 NP-hard 组合优化问题，它是传统 VRP 的变体，其中应用了容量约束。给定一组分散在某一地区的客户，需要由电动车来满足每个客户的需求。固定数量的电容电动汽车车队最初放置在仓库，并充满电。它们可以离开仓库为客户服务，并在规划期内访问充电站为电池充电。每次电动汽车访问充电站时，其电池将充满电

并最终返回仓库。目标是为电动汽车寻找路线，以便满足所有客户需求，并将车队行驶的总距离降至最低。VRP 和 CEVRP 的主要区别在于 CEVRP 考虑了电动汽车的具体特性，如车辆的自主性和电池充电的需求。电动车辆路径问题如图 4-1 所示。

CEVRP 定义在一个完全的无向图$G(V,X)$上，其中 V $=\{V_d \cup V_c \cup V_s\}$表示仓库、客户、充电站组成的节点集合。每个节点$i$都与一个数组$X_i^t=(x_i,z_i,d_i^t)$相关联，其中$x_i$和$z_i$代表节点$i$的地理坐标，而$d_i^t$是节点$i$在解码步骤 t 的剩余需求。仓库和充电站的需求被设定为 0。用步骤 t 来标示d_i和x_i，因为以顺序的方式解决问题，这两个元素可以随时间变化，X_i^t中的所有其他元素都是静态的。本章没有考虑每个节点的服务时间，因为假设它是一个常数以简化问题。所有的节点数组形成一个集合X^t，描述了解码步骤 t 时节点的局部信息。该图是完整的，每条边的权重是连接节点之间的欧氏距离。

这些节点共享一组全局变量$G^t=\{b^t,ev^t\}$,其中b^t和ev^t分别表示活动电动车的电池水平和解码步骤 t 开始时的可用电动车数量。ev^t的值最初分别设置为车队的规模，b^t的值被初始化为电动车的电池容量。所有的全局变量都可以随时间变化。在这里没有把电动车货物列为全局变量，因为它不是第 4.2.2 节中介绍的模型的输入。但确实跟踪了电动车的剩余货物，以实现掩码方案。CEVRP 的解决方案是图中的一连串节点，可以解释为电动汽车的路线。不同的电动汽车的路线被仓库分开。例如，假设节点 0 代表仓库，节点序列{0,3,2,0,4,1,0}对应两条路线：一条沿 0→3→2→0 行驶，另一条沿 0→4→1→0 行驶，意味着使用两辆电动汽车。

此外，每个客户i具有特定的交付需求q^i。所有的电动汽车都是一样的，每一辆都有最大的负载需求容量（C）和最大的电池容量（B），每辆电动汽车都不应超过这些参数。此外，他们开始（满载和充电）和结束在仓库；值得一提的是，每辆车都可以多次访问充电站，但所有客户必须被访问一次。CEVRP 目标函数的公式如下[68]：

$$\min f(x)=\sum_{i\in V,j\in V} d_{ij}\cdot x_{ij} \tag{4-1}$$

即最小化所有电动车辆的总行驶距离。受以下几个约束：

$$\sum_{j\in V,i\neq j} x_{ij}=1,\forall i\in V_c \tag{4-2}$$

$$\sum_{j\in V,i\neq j} x_{ij} \leq 1, \forall i \in V_s \tag{4-3}$$

$$\sum_{j\in V,i\neq j} x_{ij} - \sum_{j\in V,i\neq j} x_{ji}, \forall i \in V \tag{4-4}$$

$$0 \leq b^t \leq C, \forall i \in V \tag{4-5}$$

$$0 \leq y_i \leq B, \forall i \in V \tag{4-6}$$

$$x_{ij} \in \{0,1\}, \forall i \in V, \forall j \in V, i \neq j \tag{4-7}$$

其中，等式（4-2）的限制是指每个顾客只能被服务一次。另一方面，等式（4-3）指示充电站可以被访问几次。等式（4-4）通过保证在每个节点，输入弧的数量等于输出弧的数量来建立能量守恒。等式（4-5）是容量约束，其保证 EV 的负载在到达任何节点(包括存放处)时是非负的。等式（4-6）的能量约束确保电池电荷水平永远不会降到 0 以下。最后，等式（4-7）定义了二元决策变量(x_{ij})的集合，如果 EV 经过弧线(i,j)，则等于 1，否则等于 0。变量b^t和y^i分别代表电动车车辆到达节点$i \in$ V 时的剩余充电容量和剩余能量水平。尽管公式显示了明确的限制，但它们也意味着所有电动车辆必须离开和返回仓库。

4.2 基于注意力机制的 DRL 模型

4.2.1 马尔可夫决策过程模型

在本节中，从 RL 的角度来描述这个问题。假设有一个智能体通过采取一系列的动作来寻求生成 CEVRP 的解决方案。特别是，在每一步，智能体接受了当前的系统状态，并根据给定的信息采取了动作。然后，系统状态会因此而改变。这个过程不断重复，直到满足某些终止条件。本文用许多 CEVRP 实例来训练智能体，并使用奖励函数来评估智能体产生的解决方案，指导智能体进行相应的改进。

在 CEVRP 的背景下，系统状态是图信息X^t和G^t的表示。一个动作是在当前序列的末端添加（解码）一个节点。用y^t表示在步骤 t 选择的节点，用Y^t表示在步骤 t 之前形成的节点序列。假设该程序在第t_m步终止。更具体地说，在每个解码步骤 t，给定G^t、X^t和行程历史Y^t，通过$P(y^{t+1} = i|X^t, G^t, Y^t)$估计将每个节点 i 添加到序列的概率，并根

据此概率分布解码下一个要访问的节点y^{t+1}。基于y^{t+1}，使用转移函数（4-8）-（4-10）更新系统状态。

首先，活动电动车的电池电量被更新：

$$b^{t+1}=\begin{cases}b^t-f(y^t,y^{t+1}),if\ y^t\in V_c\\B-f(y^t,y^{t+1}),otherwise\end{cases} \tag{4-8}$$

其中$f(y^t,y^{t+1})$是电动车从节点y^t到节点y^{t+1}的能量消耗，B 是电池容量。

最后，每个节点可用 EV 的数量ev^t和剩余需求d_i^t（在解码步骤 t 的剩余需求）更新如下。

$$ev^{t+1}=\begin{cases}ev^t-1,if\ y^t\in V_c\\ev^t\end{cases} \tag{4-9}$$

$$d_i^{t+1}=\begin{cases}0,\ y^t=i\\d_i^t,otherwise\end{cases} \tag{4-10}$$

定义一个节点序列$Y^{tm}=\{y^0,y^1,\cdots,y^{tm}\}$，如公式（4-11）所示。一个高的奖励值对应于一个高质量的解决方案。鉴于 CEVRP 的目标是最小化车队的总行驶距离，将方程（4-11）中的第一项设定为车队的负总行驶距离，以支持短距离的解决方案。其他项是对违反问题约束的惩罚。如果解决方案Y^{tm}需要的 EV 超过给定的 EV，则相应的ev^{t_m}将是负数，这将在第二项中受到惩罚。此外，如果仓库距离充电站非常近，通过实验观察到，通过在充电站和仓库之间不断移动而不为任何客户服务，模型可能会实现较低的行驶距离。为了防止这个问题，引入了第三个条款来惩罚每一次到站，因为在 CEVRP 设置下，只在必要时访问充电站。此外，在第四项中惩罚了负的电池水平。所有其他的问题约束都在第 4.2.2 节介绍的掩码方案中得到了考虑。

$$\begin{aligned}\mathrm{r}(Y^{t_m})=&-\sum_{t=1}^{t_m}\omega(y^{t-1},y^t)+\beta_1 max\{-ev^{t_m},0\}\\&+\beta_2 S(Y^{t_m})+\beta_3\sum_{t=0}^{t_m}max\{-b^t,0\}\end{aligned} \tag{4-11}$$

其中，$\omega(y^{t-1},y^t)$是边(y^{t-1},y^t)上的行驶时间，$S(Y^{t_m})$是沿轨迹Y^{t_m}访问站点的次数，β_1、β_2和β_3是三个负常数。根据实验，方程（4-11）中说明的奖励函数可以引导 RL 智能体产生受相关约束的解决方案。然而，理论上并不能保证这些约束条件不会被违反。如果违反了，则加入下游的局部搜索启发式来进一步提高解决方案的质量[69]。

在下一节中，将详细描述 RL 方法，并解释它如何适用于 CEVRP。

4.2.2 注意力模型

本节提出了一个注意力模型来参数化上节的“概率估计向量”$P(y^{t+1}=i|X^t,G^t,Y^t)$。该模型由 3 个组件组成：以高维矢量形式表示系统状态的嵌入组件；注意力分量，用于估计每个节点的概率；长短时记忆网络（Long Short-term Memory, LSTM）解码器，以恢复行程历史。所提出的模型与 Nazari 等人[40]提出的模型之间的关键区别之一是，合并了一个图嵌入组件来合成图的局部和全局信息。模型结构如图 4-2 所示。

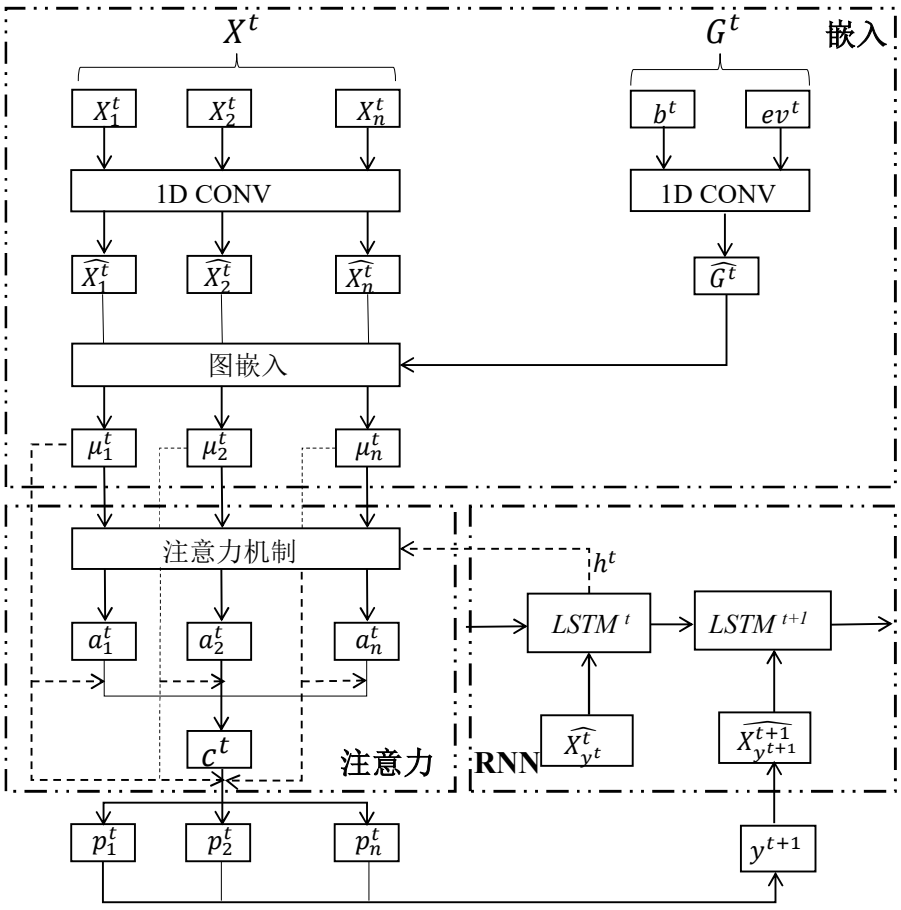

图 4-2 注意力模型

（1）图嵌入：首先将模型输入X^t和G^t映射到高维向量空间。嵌入的模型输入分别表示为$\widehat{X}^t$和$\widehat{G}^t$。更具体地说，对于节点i，其局部信息数组$X_i^t=(x_i,z_i,d_i^t)$嵌入到具有一维卷积层的ξ维向量$\widehat{X}_i^t$，嵌入层在节点之间共享。此外，还有另一个全局变量(b^t,ev^t)的一维卷积层，将它们映射到ξ维向量$\widehat{G}^t$。

然后，利用 Structure2Vec 工具合成嵌入向量[70]。为每个节点i初始化一个向量

$\mu_i^{(0)} = \widehat{X}_i^t$，然后更新$\mu_i^{(k)}$，$\forall k = 1,2,\cdots,p$，使用方程（4-12）递归。经过$p$轮递归后，网络将为每个节点$i$生成ξ维向量$\mu_i^{(p)}$，然后将嵌入向量$\mu_i^t$设置为$\mu_i^{(p)}$。

$$\mu_i^k = relu\left\{\theta_1\widehat{X}_i^t + \theta_2\widehat{G}^t + \theta_3\sum_{j\in N(i)}\mu_j^{(k-1)} + \theta_4\sum_{j\in N(i)} relu[\theta_5\omega(i,j)]\right\} \quad (4\text{-}12)$$

其中$N(i)$是通过边与节点i相连的节点集，称该集为节点i的邻域，$\omega(i,j)$表示边(i,j)上的行进时间，$\theta_1,\theta_2,\theta_3,\theta_4$和$\theta_5$是可训练变量。*relu* 是一个非线性激活函数，$relu(x) = max\{0,x\}$。

在每一轮递归中，全局信息和位置信息通过方程（4-12）的前两个项进行聚合，而不同节点和边上的信息通过最后两个求和项相互传播。最终嵌入的向量μ_i^t包含局部和全局信息，因此可以更好地表示图的复杂上下文。

（2）注意机制：基于嵌入向量μ_i^t，利用 Bahdanau 等人[71]提出的基于上下文的注意机制来计算每个节点i的访问概率。

首先计算上下文向量c^t，将整个图的状态指定为所有嵌入向量的加权和，如方程（4-13）所示。每个节点的权重在等式（4-14）和（4-15）中定义。

$$c^t = \sum_{i=0}^{|V_c|+|V_s|} a_i^t\mu_i^t \quad (4\text{-}13)$$

$$a^t = softmax(v^t) \quad (4\text{-}14)$$

$$v_i^t = \theta_v \tanh(\theta_u[\mu_i^t; h^t]) \quad (4\text{-}15)$$

其中v_i^t是矢量v^t的第i项，h^t是 LSTM 解码器的隐藏内存状态，θ_v和θ_u是可训练变量，[;]表示连接符号“;”两侧的两个矢量。tanh 是非线性激活函数，$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$是应用于向量的标准化指数函数，$\mathrm{softmax}(x) = \frac{e^x}{\sum k^{e^{x_k}}}$。然后，估计下一步访问每个节点$i$的概率，即$p_i^t$，如等式（4-16）和（4-17）所示。

$$p^t = softmax(g^t) \quad (4\text{-}16)$$

$$g_i^t = \theta_g \tanh(\theta_c[\mu_i^t; c_t]) \quad (4\text{-}17)$$

其中g_i^t是向量g^t的第i项，θ_c和θ_g是可训练变量。

（3）掩码方案：为了加快训练过程并确保解决方案的可行性，设计了几个掩码方案来排除不可行的路径。特别是，假设 EV 当前在解码步骤 t，如果节点 $j(\forall j \neq i)$满足以下条件之一，给相应的v_j^t和g_j^t分配一个很大的负数，这样计算的权重a_j^t和概率p_j^t将非常接近（如果不等于）0：

•节点 j 代表客户，其未满足的需求为零或超过 EV 的剩余货物；

•节点 j 代表客户，EV 当前电池电量b^t无法支持 EV 完成从节点i到节点 j 的行程，然后再到仓库；

•如果 EV 当前位于仓库，并且在任何客户节点都没有剩余货物，将屏蔽除仓库之外的所有节点。

（4）LSTM 解码器：使用 LSTM 来建模解码器网络。在解码步骤 t，LSTM 获取 EV 当前位置的矢量表示$\widehat{X}_{y^t}^t$以及来自先前解码步骤h^{t-1}的存储器状态输出保持关于直到步骤 t（即Y^t）的轨迹的信息的隐藏状态h^t。然后，存储器状态h^t被输入到本节前面介绍的注意力模型中[40]。

4.2.3 解码方法

给定概率 p，对于每个解码步骤 t 处的所有节点 i（由注意模型估计），智能体可以解码 CEVRP 实例的解决方案，本文考虑如下三种解码策略。

（1）贪婪解码：贪婪地选择每个步骤 t 中概率最高的节点作为下一个要访问的节点，即下一个节点$j = argmax_i p_i^t$。使用此策略，为每个实例生成一个解决方案。

（2）波束搜索：对于每种情况，同时保留具有最高总体概率的多个解决方案，并最终记录其中的最佳解决方案[72]。波束搜索可以看作是一种特殊的贪婪策略，它考虑的是解的概率而不是转移的概率。

（3）随机采样：根据p_i^t描述的概率分布对下一个要访问的节点进行采样，对于所有i，在每个解码步骤 t。可以重复此过程以获得一个实例的多个解，并以最短距离记录解决方案。

在这些策略中，贪婪解码速度最快，但由于其短视性和缺乏对解空间的探索，可能会产生较差的解决方案。随机采样和波束搜索通常可以实现更好的探索开发平衡，尽管它们可能需要更长的时间，这取决于为每个实例生成的解决方案的数量。在本文中，为了深入探索解空间，使用随机采样进行模型训练。在测试时，对这三种解码方

法进行了实现和比较。

4.2.4 策略梯度

本节实现了一个策略梯度算法来训练模型，基本思想是，并非让模型从现有算法提供的最优解中学习，而是使用前面定义的奖励函数来评估模型生成的解的质量。在每个训练迭代中，用θ表示等式（4-12）、（4-15）和（4-17）中的所有可训练变量，用π^{θ}表示相应的随机解策略。使用π^{θ}对一批随机生成的 N 个实例的解进行采样，并计算相应的奖励。基于奖励，估计了损失函数相对于每个可训练变量的梯度。然后，使用 Adam 优化器[73]更新模型中的可训练变量。

在估计梯度时，一个好的基线通常会减少训练方差，从而提高学习速度[74]。本文没有使用 Nazari 等人[40]难以实现的 A3C 方法，而是采用了 Kool 等人[74]提出的 rollout 基线。更具体地说，在第一个Λ训练步骤中，只使用模型获得的奖励的指数移动平均值。在第Λ^{th}步，将基线策略设置为在第Λ^{th}步末尾的策略。之后，每ζ次迭代评估一次基线策略。根据配对 t 检验（$\alpha = 5\%$），当且仅当当前策略明显优于单独测试集上的基线策略时，才更新基线策略。每次更新基线策略时，都会生成一个新的测试集。

将策略梯度法的关键组成部分定义如下：

（1）损失函数：目标是最小化损失函数，如方程（4-18）所示。损失函数表示使用随机策略π^{θ}采样的轨迹 Y 的负期望总回报。

$$L(\theta) = -E_{Y\sim\pi^{\theta}}[r(Y)] \tag{4-18}$$

（2）梯度估计：使用方程（4-19）估计损失函数$L(\theta)$训练变量θ的梯度。参数 N 是批次大小，$X_{[i]}$是批次中的第i个训练样本，$X_{[i]}$则是使用π^{θ}生成的相应解。此外，$BL()$表示引入的 rollout 基线，$P_{\theta}(Y_{[i]}|X_{[i]})$表示在给定训练示例$X_{[i]}$的情况下，使用随机策略 π^{θ} 生成 Y 解的概率。使用 Sutskever 等人[75]提出的概率链规则分解概率$P_{\theta}(Y_{[i]}|X_{[i]})$，如方程（4-20）所示。右侧的项$P_{\theta}(y_{[i]}^{t+1}|X_{[i]}^{t}, G_{[i]}^{t}, Y_{[i]}^{t})$可以在每个解码步骤从模型中获得。

$$\nabla_{\theta}L = \frac{1}{N}\sum_{i=1}^{N}[r(Y_{[i]}) - BL(X_{[i]})]\nabla_{\theta}logP_{\theta}(Y_{[i]}|X_{[i]}) \tag{4-19}$$

其中

$$P_\theta\left(Y_{[i]}\middle|X_{[i]}\right)=\prod_{t=0}^{|Y_{[i]}|-1}P_\theta\left(y_{[i]}^{t+1}\middle|X_{[i]}^t,G_{[i]}^t,Y_{[i]}^t\right) \tag{4-20}$$

（3）实例生成：在每个训练步骤，生成 N 个随机 CEVRP 训练实例。在每个实例中，节点均匀分布在区域$[0,1]\times[0,1]$之间。客户需求被认为是离散的，它们是从{0.05,0.10,0.15,0.20}中随机选择的，概率相等。尽管这种方法生成的实例的可行性无法保证，但根据实验，它们在大多数情况下实际上是可行的。由于 DL 模型通常对训练数据中的随机误差具有鲁棒性，所以不对这些不可行的情况进行任何调整。

将车辆规格规范化为区间[0,1]，每辆电动汽车的载货和电池容量设置为 1.0。电动汽车从 0 完全充电需要 0.25 个时间单位。对行驶一个单位距离所消耗的能量进行充电需要 0.15 个时间单位，计划范围为[0,1]。考虑在训练期间，在一个拥有 3 个充电站的地区，由 3 辆电动汽车组成的车队为 10 个客户提供服务。使用这个小实例大小来提高实例生成效率。根据数值实验，这不会影响模型性能。测试数据的生成方式与生成培训数据的方式相同，但客户、充电站和电动汽车的数量可能不同。

算法 2 总结了训练过程的伪代码。

算法 2 REINFORCE 算法

初始化网络权重θ，然后测试集 S；
for i =1,2,…,**do**
  生成 N 个随机实例$X_{[1]},X_{[2]},\cdots,X_{[N]}$
  for n =1,2,…,N **do**
    初始化步骤计算器$t_n\leftarrow 0$
    重复
      根据概率分布$P_\theta\left(y_{[n]}^{t_n+1}|X_{[n]}^{t_n},G_{[n]}^{t_n},Y_{[n]}^{t_n}\right)$选择$y_{[n]}^{t_n+1}$
      选择新状态$X_{[n]}^{t_n+1},G_{[N]}^{t_n+1},Y_{[n]}^{t_n+1}$
      $t_n\leftarrow t_n+1$
    直到满足终止条件为止
    计算奖励 r$\left(Y_{[n]}^{t_n}\right)$
  end
  if i ≤ Λ then
    BL$\left(X_{[i]}\right)\leftarrow avg\left[r\left(Y_{[1]}^{t_1},\cdots,Y_{[N]}^{t_1}\right)\right]$
  else
    BL$\left(X_{[i]}\right)\leftarrow\pi^{BL}\left(X_{[i]}\right)$
  end
  d$\theta=\frac{1}{N}\sum_{i=1}^{N}\left[r\left(Y_{[i]}\right)-BL\left(X_{[i]}\right)\right]\nabla_\theta logP_\theta\left(Y_{[i]}|X_{[i]}\right)$
  $\theta\leftarrow$ Adam$\left(\theta,\ d\theta\right)$
  if i ≤ Λ then
    初始化基线$\pi^{BL}\leftarrow\pi^\theta$
  else
    if mod $\xi=0$ and
    ONESIDETTEST$\left(\pi^\theta(S),\pi^{BL}(S)\right)<\alpha$ then
      $\pi^{BL}\leftarrow\pi^\theta$
      创建新测试集 S
    end
  end

29 **end**

4.3 实验结果与分析

4.3.1 实验设置

对于 RL 模型，本文从 Nazari 等人[40]所做的工作中调整了大部分超参数，分别使用两个单独的一维卷积层来嵌入局部信息和全局信息。所有这些信息都嵌入到一个 128 维的向量空间中。利用一个状态大小为$\xi = 128$ 的 LSTM 网络。对于 Adam 优化器，将初始步长设置为 0.001，批处理大小设置为$N = 128$。为了稳定训练，截取了梯度d_θ，使其范数不超过 2.0。关于 rollout 基线，在前 1000 个训练步骤中使用移动指数平均基线，并在之后每 100 个训练步骤评估基线策略。在奖励函数中，仓库访问和充电站访问的惩罚因子和负极电量分别设置为 1.0,0.3 和 100，训练模型 10000 次迭代。在实验环境方面，本章所有实验使用同一配置进行：Inter（R）Core（TM）i7-11700 CPU，NVIDIA GeForce RTX 3060 Ti GRU 的计算机上进行实验，代码用 python 实现。

在训练模型时，以随机的方式对解进行抽样，以使模型遇到的可能情况多样化。在测试时，考虑了所有三种解码方法，并比较了它们的性能。在实现随机解码进行测试时，为每个实例采样 100 个解，并记录总距离最短的解。对于波束搜索，同时保留 3 个解，并记录总体概率最高的一个解。

4.3.2 比较分析

本文比较了两种方法的性能：由 Schneider 等人[76]开发的 VNS/TS 启发式，以及提出的 RL 方法。将这些解决方案方法应用于六个不同的场景：例如“C5-S2-EV2”是指 5 个客户，2 个充电站，2 辆电动汽车的场景。对于每个场景，解决了 100 个以相同方式生成训练数据的创建的实例，并在表 4-1 中记录 EV 车队的平均总行驶距离（Dis.），以及这些算法实现的最小距离之间的差距（Gap）。表 4-2 中记录了以秒为单位的 100 个实例的平均解决时间。本文只记录能在 15 分钟内成功解决一个实例的算法的结果。

表 4-1 四种方法的平均总行驶距离比较

实例	VNS/TS[76]		RL（Greedy）		RL（Beam）[72]		RL（Stochastic）	
	Dis.	Gap	Dis.	Gap	Dis.	Gap	Dis.	Gap
C5-S2-EV2	2.33	0%	2.64	13.30%	2.64	13.30%	2.51	7.73%
C10-S3-EV3	3.64	0%	4.42	21.42%	4.38	20.32%	4.04	10.99%
C20-S3-EV3	5.34	0%	7.27	36.14%	7.48	40.07%	6.41	20.04%
C30-S4-EV4	6.87	0%	9.76	42.07%	10.58	54.00%	8.46	23.14%
C40-S5-EV5	-	-	12.70	13.70%	14.72	31.78%	11.17	0%
C50-S6-EV6	-	-	16.46	14.94%	18.64	30.17%	14.32	0%

在三种 RL 实现中，随机解码方法虽然比波束搜索和贪婪解码更耗时，但总能得到

质量最好的解。这一发现与 Barrett 等人[34]提出的结果一致，即学习直接产生单一最优解的策略通常是不切实际的。相反，用随机策略探索解空间通常会得到比单一“最佳猜测”更好的解。

表 4-2 四种方法的平均求解时间比较

实例	VNS/TS[76]（s）	RL（Greedy）（s）	RL（Beam）[72]（s）	RL（Stochastic）（s）
C5-S2-EV2	1.32	0.17	0.20	2.88
C10-S3-EV3	10.37	0.35	0.40	7.63
C20-S3-EV3	168.86	0.62	0.71	19.40
C30-S4-EV4	536.80	1.06	1.17	43.06
C40-S5-EV5	-	1.69	1.86	70.26
C50-S6-EV6	-	2.31	2.61	107.96

在小实例上，本文所提出的方法能有效地找到可行的解决方案，但解的质量不如 VNS/TS 启发式。对于“C5-S2-EV2”和“C10-S3-EV3”场景，RL（随机抽样）实现的最优性差距分别为 7.73%和 10.99%，而 VNS/TS 启发式在大多数情况下可以将问题求解到最优。然而，RL 模型比和 VNS/TS 启发式具有更好的可扩展性和泛化能力。在“C20-S3-EV3”和“C30-S4-EV4”场景上，VNS/TS 启发式在解决方案质量方面优于 RL 模型，但所花费的解决时间是 RL 模型的 7-10 倍。对于拥有 40 个或更多客户的场景，RL 模型是唯一能够在 15 分钟内解决 CEVRP 的算法。事实上，RL 模型解决 50 个客户的实例平均只花费 1.8 分钟左右。

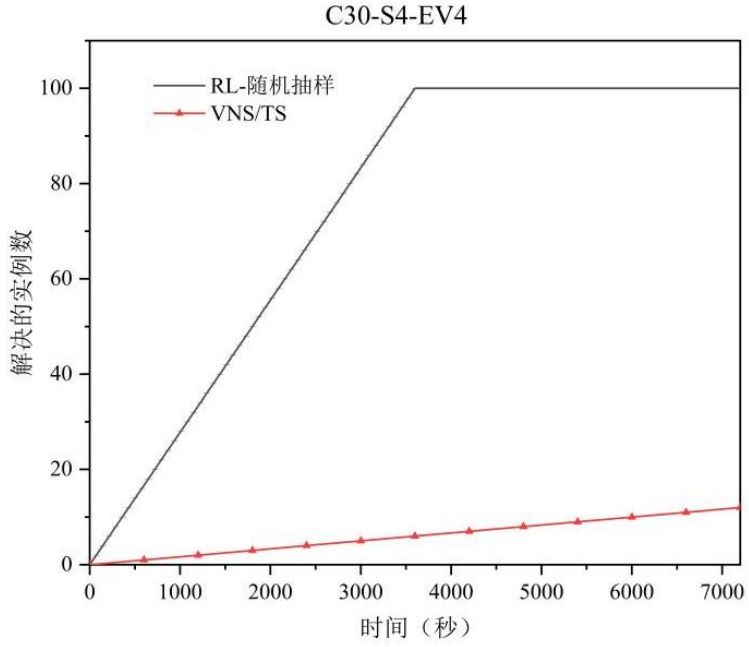

图 4-3 DRL 模型和 VNS/TS 启发式求解实例的数量比较图

然后考虑取消 15 分钟的求解时间限制，并比较 VNS/TS 启发式和 RL 模型(随机抽

样)在两小时内可以解决的实例数量，结果如图 4-3 所示和图 4-4 所示。对于“C30-S4-EV4”场景，RL 模型在大约 60 分钟内解决了所有 100 个给定实例，而 VNS/TS 启发式仅在 2 小时内解决了 12 个实例。对于场景“C50-S6-EV6”，RL 智能体解决的实例大约比 VNS/TS 启发式多 1500%。考虑到实际商用电动汽车车队的规模，RL 智能体是适用于大规模动态调度的唯一方法。

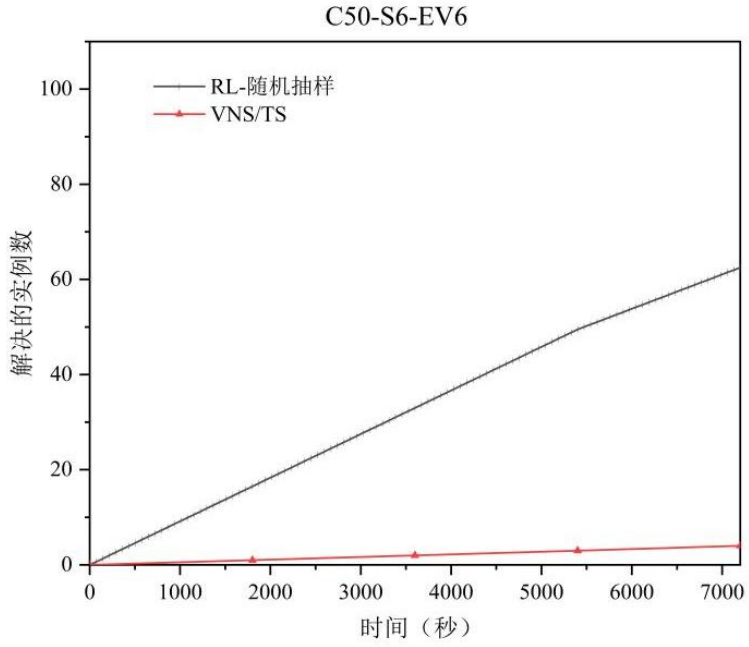

图 4-4 DRL 模型和 VNS/TS 启发式求解实例的数量比较图

4.3.3 算法分析

本节将对所提出的方法进行详细分析：图 4-5 分别给出了随机抽样和 VNS/TS 启发式下 RL 智能体在两个实例上生成的路线，其中（a）和（b）是实例一，（c）和（d）是实例二，与客户、充电站和仓库对应的节点以不同的形状标示。

进一步对实例一的仓库位置进行敏感性分析。图 4-6 显示了当改变仓库和充电站的位置时，在步骤 0 中计算的概率分布。当仓库位于节点 0 时，电动汽车最可能访问距离仓库最近的客户 8。将仓库移动到节点 11 或 12 时，由于节点 3 更接近仓库，访问概率增加。出于类似的原因，将仓库移动到节点 11 时，节点 6 被分配了一个小概率。RL 智能体展示了其考虑位置信息优化电动汽车车队路线的能力。产生的客户序列，虽然不一定是最优的，但通常是高质量的。

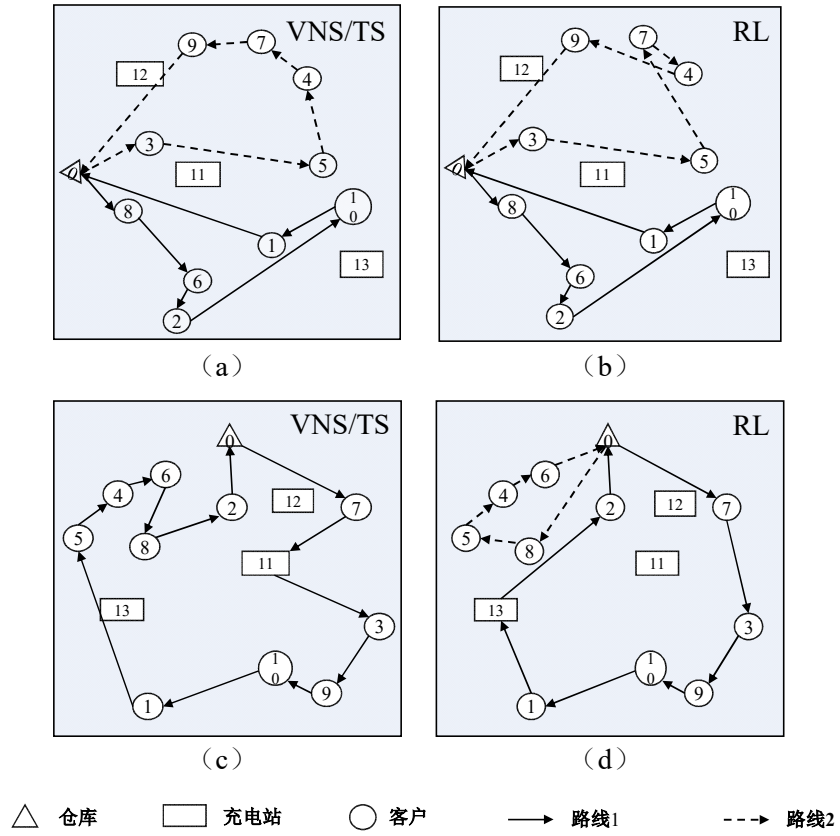

图 4-5 VNS/TS 启发式和 RL 模型（随机抽样）生成两个实例的样本路径

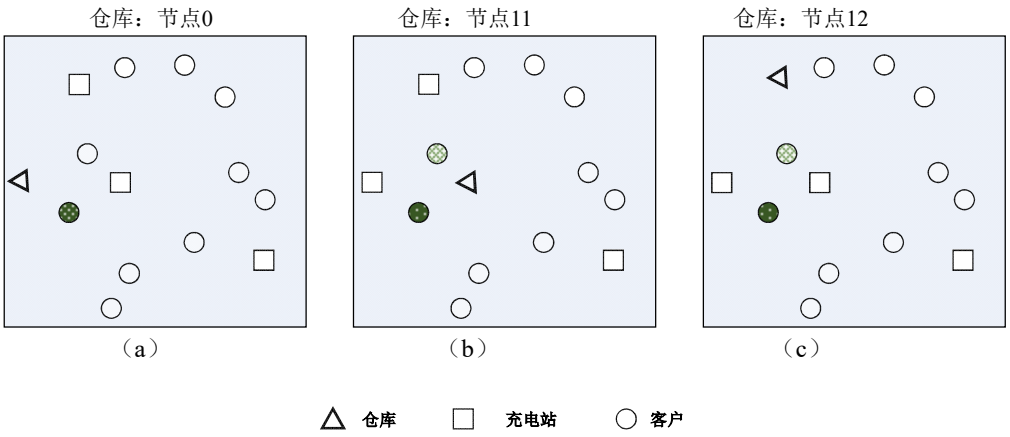

图 4-6 实例一不同节点位置下步骤 0 的概率分布可视化

然而，研究发现 RL 智能体有时是短视的，特别是在充电决策方面。它通常无法在旅途早期发现充电的机会。以图 4-5 中的实例二为例，VNS/TS 启发式仅利用一个 EV 服务所有客户，而 RL 智能体需要两辆 EV。原因是，在 VNS/TS 启发式生成的解决方案中，电动汽车在服务客户 7 后立即给电池充电。因此，电动汽车有足够的电量完成剩下的旅程，并设法在不违反任何时间窗口的情况下服务所有的客户。由 RL 智能体生

成的路线 1 以非常相似的顺序穿过客户，而不会绕道到 11 站。当 RL 智能体最终意识到电动汽车的能量不足时，它在服务客户 1 后将电动汽车发送到 13 站。这个弯路加上 13 号站的充电时间，此时前往 8 号客户已经不能满足其容量需求。因此 RL 智能体需要另一辆 EV。另外，研究发现，晚充电的缺点部分是由于 CEVRP 的完全充电假设，即电动汽车充电越晚，需要的充电时间越长。

综上所述，提出的 RL 模型能够捕获嵌入在给定图中的结构，并结合位置和时间信息为路径决策提供信息，客户得到的序列通常是高质量的。在充电方面，RL 智能体主要根据电动汽车的电池电量做出充电决策。因此，它可以确保电动汽车在能量不足时仍能充电，但可能会错过一些充电机会，特别是在早期阶段。可以通过开发和培训一个单独的充电决策模型来进行改进。此外，放宽 CEVRP 的完全充电假设也是未来研究的一个有趣的方向。

4.4 本章小结

在本章中，开发了一个用于求解 CEVRP 的强化学习框架。该算法展示了强大的可扩展性，它能够解决现有方法无法解决的非常大的实例，分析表明，所提出的模型能够快速捕获图中嵌入的重要信息，进而有效地为问题提供相对较好的可行解。这些可行的解决方案虽然不是最优的，但可以用于支持大规模实时电动汽车运行。此外，提出的模型可以潜在地扩展到 CEVRP 的其他变体。实践者可以根据自己的操作约束和目标，通过轻微地调整屏蔽方案和奖励函数来扩展所提出的方法，这比调整其他精确或元启发式算法容易得多，这些算法通常需要特殊的假设和领域知识。

5 总结和展望

物流行业正在迅猛发展，它不仅成为我国人力和资源之外的利润来源，而且已经成为支撑我国综合实力的重要支柱。随着人工智能和硬件算力的不断提升，研发一种快速求解调度优化的高效算法已经成为亟待解决的问题。

本文在已有工作的基础上主要内容如下：

（1）提出了一种基于 DRL 的神经构造启发式方法，以解决异构车辆不同容量限制问题，其目的是最小化车队中车辆的最长行驶时间或总时间。利用车辆选择解码器进行异构车辆的选择，节点选择解码器负责路线构建，该解码器通过在每一步自动选择车辆和该车辆的节点来学习构建解决方案。并采用 REINFORCE 算法进行训练，从而提高模型的求解性能。在选择策略时，采用贪婪动作选择策略以及采样动作选择策略。实验结果表明，DRL 方法可以有效地处理异构车辆并实现更低的最优性差距，计算时间稍长。此外，所提出的方法优于大多数比较启发式方法，并且与基线启发式方法 SISR 相比，具有令人满意的计算时间。

（2）提出一个端到端的 DRL 框架来解决电动汽车路径规划，为了对电动汽车的运营进行建模，开发了一个结合了指针网络和图嵌入层的注意力模型，为解决 CEVRP 的随机策略提供参数。特别是在仅考虑节点信息的框架中，加入图嵌入组件以及全局信息，实现图的局部信息和整体信息的综合定义。然后使用带有 rollout 基线的 REINFORCE 策略梯度方法来训练该模型，通过奖励函数来评估智能体产生的解决方案，指导智能体进行相应的改进。在选择策略时考虑三种解码策略，分别是贪婪解码、波束搜索以及随机采样。为了深入探索解空间，本文中使用随机抽样进行模型训练，在测试时对这三种解码方法进行实现和比较。研究表明，所提出的模型能够有效地解决当前现有方法无法解决的大规模 CEVRP 实例。

未来的研究方向：应用基于 DRL 的方法解决车辆路径问题的研究方向虽然具有良好的潜力，但仍处于早期阶段。本文提出的研究已经解决了车辆路径中两个变体，然而现实应用中仍然具有一些其它具有挑战和实际的约束需要考虑：

（1）时间窗口限制

目前的深度神经模型，大多强调经典的问题特定特征，如位置坐标和客户节点的需求，很少研究时间维度的特征。然而具有时间窗口约束的车辆路径问题（vehicle

routing problems with time window, VRPTW）在每个客户额外分配访问时间间隔，在分销管理等行业中广泛应用。VRPTW 作为 CVRP 的延伸，限制了车辆必须在其时间窗口内为客户服务，这进一步增加了 DRL 捕捉和学习问题特征和约束条件的难度。因此，未来更需要继续探索具有时间窗口约束的车辆路径问题。

（2）动态交通和客户需求

本文所提出的工作重点是学习静态交通条件和客户需求下的问题的底层模式，从而提前产生高质量的解。然而现实情况中经常发生的交通拥堵可能会造成车辆到达的延迟，进而使得行程时间不可预测。此外，客户的请求可能会随机出现，在车辆配送过程中，未被访问过的客户需求可能会发生改变，导致车辆根据已有调度防范执行时出现车载物资不满足客户物资需求量，或者按当前调度方案执行时出现时效性差等情况。因此，需求量动态变化的车辆路径规划仍是未来需要重点关注的问题。

参考文献

[1] 牛鹏飞, 王晓峰, 芦磊, 等. 强化学习在车辆路径问题中的研究综述[J]. 计算机工程与应用, 2022, 58(1): 41-55.

[2] Golden B, Assad A, Levy L, et al. The fleet size and mix vehicle routing problem[J]. Computers Operations Research, 1984, 11(1): 49-66.

[3] Koç Ç, Bektaş T, Jabali O, et al. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows[J]. Computers Operations Research, 2015, 64: 11-27.

[4] Wang X, Poikonen S, Golden B. The vehicle routing problem with drones: several worst-case results[J]. Optimization Letters, 2017, 11: 679-697.

[5] 薛星群, 王旭坪, 韩涛, 等. 考虑通行约束和运力限制的灾后应急物资联合调度优化研究[J]. 中国管理科学, 2020, 28(3): 21-30.

[6] Desrochers M, Desrosiers J, Solomon M. A new optimization algorithm for the vehicle routing problem with time windows[J]. Operations Research, 1992, 40(2): 342-354.

[7] Dantzig G B, Ramser J H. The truck dispatching problem[J]. Management Science, 1959, 6(1): 80-91.

[8] Zhang Z, Qin H, Li Y. Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(5): 1987-2001.

[9] Dorling K, Heinrichs J, Messier G G, et al. Vehicle routing problems for drone delivery[J]. IEEE Transactions on Systems, Man, Cybernetics: Systems, 2016, 47(1): 70-85.

[10] Yaman H. Formulations and valid inequalities for the heterogeneous vehicle routing problem[J]. Mathematical Programming, 2006, 106: 365-390.

[11] Baldacci R, Mingozzi A. A unified exact method for solving different classes of vehicle routing problems[J]. Mathematical Programming, 2009, 120: 347-380.

[12] Pessoa A, Uchoa E, Poggi De Aragão M. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem[J]. Networks: An International Journal, 2009, 54(4): 167-177.

[13] Baldacci R, Bartolini E, Mingozzi A, et al. An exact solution framework for a broad class of vehicle routing problems[J]. Computational Management Science, 2010, 7(3): 229.

[14] Jabali O, Gendreau M, Laporte G. A continuous approximation model for the fleet composition problem[J]. Transportation Research Part B: Methodological, 2012, 46(10): 1591-1606.

[15] Valle C A, Martinez L C, Da Cunha A S, et al. Heuristic and exact algorithms for a min‐max selective vehicle routing problem[J]. Computers Operations Research, 2011, 38(7): 1054-1065.

[16] Bianchessi N, Corberán Á, Plana I, et al. The min-max close-enough arc routing problem[J]. European Journal of Operational Research, 2022, 300(3): 837-851.

[17] Paraskevopoulos D C, Laporte G, Repoussis P P, et al. Resource constrained routing and scheduling: Review and research prospects[J]. European Journal of Operational Research, 2017, 263(3): 737-754.

[18] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.

[19] Glover F.Tabu search-part I[J]. ORSA Journal on Computing, 1989, 1(3): 190-206.

[20] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems[C]//Principles and Practice of Constraint Programming—CP98: 4th International Conference, CP98 Pisa, Italy, October 26–30, 1998 Proceedings 4. Springer Berlin Heidelberg, 1998: 417-431.

[21] Grangier P, Gendreau M, Lehuédé F, et al. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization[J]. European Journal of Operational Research, 2016, 254(1): 80-91.

[22] Žulj I, Kramer S, Schneider M. A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem[J]. European Journal of Operational Research, 2018, 264(2): 653-664.

[23] Zirour M. Vehicle routing problem: models and solutions[J]. Journal of Quality Measurement Analysis, 2008, 4(1): 205-218.

[24] Gheysens F, Golden B, Assad A. A comparison of techniques for solving the fleet size and mix vehicle routing problem[J]. Operations-Research-Spektrum, 1984, 6: 207-216.

[25] Ochi L S, Vianna D S, Drummond L M, et al. A parallel evolutionary algorithm for the vehicle

routing problem with heterogeneous fleet[J]. Future Generation Computer Systems, 1998, 14(5-6): 285-292.

[26] Prins C. Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case[J]. Journal of Mathematical Modelling Algorithms, 2002, 1(2): 135-150.

[27] Vidal T, Crainic T G, Gendreau M, et al. A unified solution framework for multi-attribute vehicle routing problems[J]. European Journal of Operational Research, 2014, 234(3): 658-673.

[28] Feng L, Zhou L, Gupta A, et al. Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking[J]. IEEE Transactions on Cybernetics, 2019, 51(6): 3171-3184.

[29] Goeke D. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles[J]. European Journal of Operational Research, 2019, 278(3): 821-836.

[30] Yilmaz Y, Kalayci C B. Variable Neighborhood Search Algorithms to Solve the Electric Vehicle Routing Problem with Simultaneous Pickup and Delivery[J]. Mathematics, 2022, 10(17): 3108.

[31] 吴廷映,孙灏.考虑载重影响耗电率的电动车车辆路径问题[J].控制与决策,2023,38(02):483-491.DOI:10.13195/j.kzyjc.2021.1050.

[32] Ozbaygin G, Savelsbergh M. An iterative re-optimization framework for the dynamic vehicle routing problem with roaming delivery locations[J]. Transportation Research Part B: Methodological, 2019, 128: 207-235.

[33] Duman E N, Taş D, Çatay B. Branch-and-price-and-cut methods for the electric vehicle routing problem with time windows[J]. International Journal of Production Research, 2022, 60(17): 5332-5353.

[34] Ahmadi S, Tack G, Harabor D, et al. Vehicle Dynamics in Pickup-And-Delivery Problems Using Electric Vehicles[C]//27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[35] Grandinetti L, Guerriero F, Pezzella F, et al. A pick-up and delivery problem with time windows by electric vehicles[J]. International Journal of Productivity Quality Management, 2016, 18(2-3): 403-423.

[36] Manchanda S, Mittal A, Dhawan A, et al. Learning heuristics over large graphs via deep reinforcement learning[J]. ArXiv Preprint ArXiv:1903.03332, 2019.

[37] Li Z, Chen Q, Koltun V. Combinatorial optimization with graph convolutional networks and guided tree search[J]. Advances in Neural Information Processing Systems, 2018, 31.

[38] Barrett T, Clements W, Foerster J, et al. Exploratory combinatorial optimization with reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(4): 3243-3250.

[39] Bello I, Pham H, Le Q V, et al. Neural combinatorial optimization with reinforcement learning[J]. ArXiv Preprint ArXiv:1611.09940, 2016.

[40] Nazari M, Oroojlooy A, Snyder L, et al. Reinforcement learning for solving the vehicle routing problem[J]. Advances in Neural Information Processing Systems, 2018, 31.

[41] Voulodimos A, Doulamis N, Doulamis A, et al. Deep learning for computer vision: A brief review[J]. Computational Intelligence Neuroscience, 2018, 2018.

[42] Chien J T. Deep Bayesian natural language processing[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts. 2019: 25-30.

[43] Goyal P, Pandey S, Jain K. Deep learning for natural language processing[J]. New York: Apress, 2018.

[44] Ma Y, Li J, Cao Z, et al. Learning to iteratively solve routing problems with dual-aspect collaborative transformer[J]. Advances in Neural Information Processing Systems, 2021, 34: 11096-11107.

[45] Bengio Y, Lodi A, Prouvost A. Machine learning for combinatorial optimization: a methodological tour d’horizon[J]. European Journal of Operational Research, 2021, 290(2): 405-421.

[46] Falkner J K, Thyssens D, Schmidt-Thieme L. Large Neighborhood Search based on Neural Construction Heuristics[J]. ArXiv Preprint ArXiv:2205.00772, 2022.

[47] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30.

[48] Kool W, Van Hoof H, Welling M. Attention, learn to solve routing problems![J]. ArXiv Preprint ArXiv:1803.08475, 2018.

[49] Joshi C K, Laurent T, Bresson X. On learning paradigms for the travelling salesman problem[J]. ArXiv Preprint ArXiv:1910.07210, 2019.

[50] Peng B, Wang J, Zhang Z. A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems[C]//Artificial Intelligence Algorithms and Applications: 11th

International Symposium, ISICA 2019, Guangzhou, China, November 16–17, 2019, Revised Selected Papers 11. Springer Singapore, 2020: 636-650.

[51] Wu Y, Song W, Cao Z, et al. Learning improvement heuristics for solving routing problems[J]. IEEE Transactions on Neural Networks Learning Systems, 2021, 33(9): 5057-5069.

[52] 孙志军, 薛磊, 许阳明, 等. 深度学习研究综述[J]. 计算机应用研究, 2012, 29 (08): 2806-2810.

[53] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[54] Hinton G E, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7): 1527-1554.

[55] Zhong Z, Lin Z Q, Bidart R, et al. Squeeze-and-attention networks for semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 13065-13074.

[56] Ulutan O, Iftekhar A S M, Manjunath B S. Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 13617-13626.

[57] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11534-11542.

[58] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(01): 1-27.

[59] Nguyen T T, Nguyen N D, Nahavandi S. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications[J]. IEEE Transactions on Cybernetics, 2020, 50(9): 3826-3839.

[60] 谢颖. 交通控制子区动态划分和信号协调优化控制[D].广西工学院,2011.

[61] Vinyals O, Bengio S, Kudlur M. Order matters: Sequence to sequence for sets[J]. ArXiv Preprint ArXiv:1511.06391, 2015.

[62] Ho J, Kalchbrenner N, Weissenborn D, et al. Axial attention in multidimensional transformers[J]. ArXiv Preprint ArXiv:1912.12180, 2019.

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

[64] Xu Z, Cai Y. Variable neighborhood search for consistent vehicle routing problem[J]. Expert Systems with Applications, 2018, 113: 66-76.

[65] Palma-Blanco A, González E R, Paternina-Arboleda C D. A two-pheromone trail ant colony system approach for the heterogeneous vehicle routing problem with time windows, multiple products and product incompatibility[C]//Computational Logistics: 10th International Conference, ICCL 2019, Barranquilla, Colombia, September 30–October 2, 2019, Proceedings 10. Springer International Publishing, 2019: 248-264.

[66] Matthopoulos P-P, Sofianopoulou S. A firefly algorithm for the heterogeneous fixed fleet vehicle routing problem[J]. International Journal of Industrial Systems Engineering, 2019, 33(2): 204-224.

[67] Pessoa A, Sadykov R, Uchoa E, et al. A generic exact solver for vehicle routing and related problems[J]. Mathematical Programming, 2020, 183: 483-523.

[68] Mavrovouniotis M, Menelaou C, Timotheou S, et al. A benchmark test suite for the electric capacitated vehicle routing problem[C]//2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2020: 1-8.

[69] Zhao J, Mao M, Zhao X, et al. A hybrid of deep reinforcement learning and local search for the vehicle routing problems[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(11): 7208-7218.

[70] Dai H, Dai B, Song L. Discriminative embeddings of latent variable models for structured data[C]//International Conference on Machine Learning. PMLR, 2016: 2702-2711.

[71] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. ArXiv Preprint ArXiv:1409.0473, 2014.

[72] Neubig G. Neural machine translation and sequence-to-sequence models: A tutorial[J]. ArXiv Preprint ArXiv:1703.01619, 2017.

[73] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. ArXiv Preprint ArXiv:1412.6980, 2014.

[74] Xiong C, Zhong V, Socher R. Dynamic coattention networks for question answering[J]. ArXiv Preprint ArXiv:1611.01604, 2016.

[75] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in

Neural Information Processing Systems, 2014, 27.

[76] Schneider M, Stenger A, Goeke D. The electric vehicle-routing problem with time windows and recharging stations[J]. Transportation Science, 2014, 48(4): 500-520.