控制与决策
Control and Decision

基于深度强化学习的多配送中心车辆路径规划

王万良, 陈浩立, 李国庆, 冷龙龙, 赵燕伟

引用本文:
王万良,陈浩立,李国庆,冷龙龙,赵燕伟. 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.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.1381

基于深度强化学习的多配送中心车辆路径规划

王万良†, 陈浩立, 李国庆, 冷龙龙, 赵燕伟

(浙江工业大学 计算机科学与技术学院，杭州 310023)

摘 要: 多配送中心车辆路径规划(multi-depot vehicle routing problem, MDVRP)是现阶段供应链应用较为广泛的问题模型,现有算法多采用启发式方法,其求解速度慢且无法保证解的质量,因此研究快速且有效的求解算法具有重要的学术意义和应用价值. 以最小化总车辆路径距离为目标,提出一种基于多智能体深度强化学习的求解模型. 首先,定义多配送中心车辆路径问题的多智能体强化学习形式,包括状态、动作、回报以及状态转移函数,使模型能够利用多智能体强化学习训练;然后通过对MDVRP的节点邻居及遮掩机制的定义,基于注意力机制设计由多个智能体网络构成的策略网络模型,并利用策略梯度算法进行训练以获得能够快速求解的模型;接着,利用2-opt局部搜索策略和采样搜索策略改进解的质量;最后,通过对不同规模问题仿真实验以及与其他算法进行对比,验证所提出的多智能体深度强化学习模型及其与搜索策略的结合能够快速获得高质量的解.

关键词: 多配送中心车辆路径规划；强化学习；多智能体；注意力机制；策略梯度；局部搜索

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.1381

引用格式: 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.

Deep reinforcement learning for multi-depot vehicle routing problem

WANG Wan-liang†, CHEN Hao-li, LI Guo-qing, LENG Long-long, ZHAO Yan-wei

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: The multi-depot vehicle routing problem (MDVRP) is widely used in the supply chain at present. Most of the existing algorithms use heuristic methods, which are slow to solve the problem and cannot guarantee the quality of the solution. It is of great academic significance and application value to study a fast and high-quality algorithm to solve the problem. With the goal of minimizing the total vehicle routing distance, a multi-agent deep reinforcement learning model is proposed. Firstly, the form of multi-agent reinforcement learning for the multi-depot vehicle routing problem is defined, including state, action, reward, and transition function, so that the model can be trained by multi-agent reinforcement learning. Through the definition of node neighbor and the masking mechanism of the MDVRP, a policy network composed of multi-agent networks based on the attention mechanism is designed. And the policy gradient algorithm is used to train the model. Then, the 2-opt local search strategy and the sampling search strategy are used to improve the solution. Finally through the comparison of the simulation experiments of different scale problems with other algorithms, it is verified that the proposed multi-agent deep reinforcement learning model and its combination with the search strategy can obtain high-quality solutions within a short period.

Keywords: mult-depot vehicle routing problem; reinforcement learning; multi agent; attention mechanism; policy gradient; local search

0 引 言

车辆路径问题(vehicle routing problem, VRP)于1959年首次提出,是物流配送行业的核心问题. 随后,针对不同的实际需求,国内外学者提出了不同的VRP变体. 其中,多配送中心车辆路径问题(multi-depot vehicle routing problem, MDVRP)是更加复杂的物流配送问题,由Gillett等[1]于1976年首次提出,其任务是在车辆最大容量、顾客需求服务等多个约束条件下,以总配送成本最小为目标,设计最优调度方案实现多个配送中心为多个顾客点配送服务. 由于MDVRP是一个NP难问题,需要花费十分昂贵的时间成本才能获得该问题的最优解,如何更快速地获得

收稿日期: 2021-08-07；**录用日期:** 2022-01-28.
基金项目: 国家自然科学基金项目(61873240).
†通讯作者. E-mail: zjutwwl@zjut.edu.cn.

高质量的解成为该领域的研究重点.

在已有研究中,少部分学者采用精确算法进行求解,例如Bettinelli等[2]设计了一种分支定价算法求解带有时间窗约束的多车型MDVRP. 但精确算法在求解MDVRP等复杂问题时效果并不理想,因此大部分文献采取启发式方法进行求解. 文献[3-4]采用不同的聚类方法将MDVRP分解为多个单配送中心VRP进行求解; Ho等[5]提出一种融合邻域搜索与遗传算法的混合启发式方法,采用节约算法[6]和最近邻启发式(nearest neighbor heuristic, NNH)生成初始种群,并在遗传操作中采用迭代交换进程(iterated swap procedure, ISP). Bezerra等[7]采用扫描算法对顾客点聚类并生成初始解,设计了一种生成变邻域搜索方法改进解的质量. 王征等[8]提出一种改进的变邻域搜索算法求解带有时间窗约束的MDVRP. 许维胜等[9]利用改进的最优切割算法MDVRP-Split将顾客分配给配送中心以求解两级车辆路径问题. 曾正洋等[10]为了解决应急物流问题,建立累计时间式MDVRP模型,并提出一种多起点变邻域下降法进行求解. 周鲜成等[11]构建多配送中心绿色车辆路径问题(MDGVRP)模型,并设计了一种改进的蚁群算法进行求解.

深度强化学习(deep reinforcement learning, DRL)是深度学习与强化学习的结合,其利用深度学习强大的表征能力拟合强化学习中智能体的动作价值函数或动作策略函数,有效地解决了动作空间及状态空间过大情况下动作价值和状态价值的存储问题. 随着近几年深度强化学习的兴起,越来越多的国内外学者开始研究其在求解组合优化问题(combinatorial optimization problems, COP)中的应用,其中车辆路径问题是典型的组合优化问题. Vinyals等[12]提出了pointer network神经网络模型求解旅行商问题(travelling salesman problem, TSP),首次以端到端的求解方式将深度学习应用到车辆路径问题中. 神经网络监督式学习需要TSP问题的最优解作为训练标签,而通常很难获得一个TSP问题的最优解,为此Bello等[13]采用强化学习代替监督式训练方式,这也是首次利用深度强化学习直接求解组合优化问题. Nazari等[14]认为车辆路径问题的解与点的输入顺序无关,因此将pointer network中的编码器部分从循环神经网络替换为线性嵌入层. Kool等[15]在编码器的设计上采用Transformer结构[16]提出了一种基于图注意力机制的网络模型,并利用带有基准的策略梯度算法训练网络,实验中利用该模型有效地求解了包括VRP在内的多个组合优化问题. 文献[15]中的模型是目前求解COP较为高效的深度强化学习方法. 以上所提到的方法均采用编码器-解码器结构,另有部分学者采用图神经网络模型进行求解. 例如Li等[17]利用图卷积网络估计图中每个顶点属于最优解的概率,并通过树搜索求解多个组合优化问题. Nowak等[18]利用图卷积网络输出图中每条边被选择的概率,并采用波次搜索获得TSP最优解,通过LKH3[19]获得训练标签,用监督学习的方式训练网络. 还有学者将启发式方法与深度强化学习相结合,通过DRL训练启发式模型. 例如Chen等[20]提出了基于深度强化学习的局部重写启发式方法NeuRewriter,利用Q-Actor-Critic方式训练网络模型. Lu等[21]提出一个基于学习的改进方法,通过网络模型输出改进算子池中选择的改进算子以及重构算子池中的重构算子. Costa等[22]提出了基于学习的2-opt局部搜索方法,通过网络模型选择2-opt的操作节点. Wu等[23]将深度强化学习方法与启发式方法相结合,利用DRL学习车辆路径问题的启发式规则,该方法的实验结果优于端到端的DRL方法.

综上所述,目前对MDVRP的研究多集中于启发式方法,而随着DRL在COP中的研究发展,该领域仍存在以下局限性:

1) 启发式方法通常采用“先分组后规划”的思想,不同分组各自规划,导致分组之间整体关联性的缺失;

2) 启发式方法中分组的优劣通常决定了整体规划的优劣,而分组规则的制定需要大量专家邻域知识,人为的分组规则制定很难达到最优效果;

3) 目前DRL方法在COP中的研究主要集中在利用单个智能体学习与环境之间的交互求解TSP、VRP等问题,而关于求解MDVRP问题的研究相对缺乏.

针对现有研究的局限性,本文根据MDVRP特点,提出一种基于多智能体深度强化学习的MDVRP求解模型MADRL-model,利用深度学习的注意力网络提取得到更高维度的特征表示,并通过多智能体之间学习相互合作行为提高模型决策性能. 首先定义MDVRP的多智能体强化学习形式,根据MDVRP问题特性设计策略网络中的邻居结构和解码器的遮掩机制,并设计一种新的基于多智能体强化学习的策略网络框架. 在不分组的情况下从整体上进行车辆路径规划,能够以端到端的形式快速有效地求解MDVRP. 针对模型求解结果中出现的问题,采用不同的局部搜索策略对解进行改进. 最后,通过对不同规

模问题仿真实验以及与其他算法进行对比,验证所提出的多智能体深度强化学习模型及其与搜索策略的结合能够快速获得高质量的解.

1　问题与模型

1.1　问题描述

MDVRP描述如下:存在多个配送中心为多个顾客点配送所需货物,每个配送中心分配一定数量的车辆,车辆运输最大容量以及配送中心位置坐标已知;每个顾客点的需求量以及位置坐标已知;目标是通过最优车辆路径规划实现总路径最小化. 为了便于分析和研究,对该问题作出以下假设:

1) 所有车辆车型相同;

2) 所有顾客需求均小于车辆最大容量;

3) 每个顾客点有且仅有一个配送中心为其服务;

4) 每辆车均从一个配送中心出发且最终返回同一个配送中心;

5) 配送中心之间不存在运输路线.

图1为MDVRP的一个简单示例.

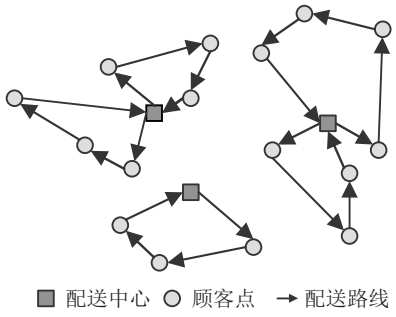


图 1　MDVRP示例图

1.2　MDVRP数学模型

假设有D个配送中心,配送中心集合为$\mathrm{Dep}=\{1,2,\dots,D\}$;有$C$个顾客点,顾客集合为$\mathrm{Cus}=\{D+1,D+2,\dots,D+C\}$,顾客点的需求为$\lambda_i, i\in\mathrm{Cus}$;所有节点的集合为$V=\mathrm{Dep}\bigcup\mathrm{Cus}$;车辆集合为$K=\{K_d\}, d=1,2,\dots,D$,$K_d$为配送中心$d$的车辆集合,总车辆数为$L=|K|$;车辆的最大容量为$Q$,对于$\forall i\in V, \lambda_i\leqslant Q$;所有配送中心与顾客点之间的距离采用欧氏距离$\mathrm{Dis}(i,j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}, i,j\in V$. MDVRP数学模型如下:

$$\min\sum_{d\in\mathrm{Dep}}\sum_{k\in K_d}\sum_{i\in V}\sum_{j\in V}\mathrm{Dis}(i,j)\times x_{ijk}. \tag{1}$$

$$\text{s.t.}\sum_{i\in V}x_{ijk}=\sum_{i\in V}x_{jik},\ \forall j\in\mathrm{Cus},\ k\in K; \tag{2}$$

$$\sum_{k\in K}\sum_{j\in V}x_{ijk}=1,\ \forall i\in\mathrm{Cus}; \tag{3}$$

$$\sum_{i\in\mathrm{Cus}}\lambda_i\sum_{j\in V}x_{ijk}\leqslant Q,\ \forall k\in K; \tag{4}$$

$$\sum_{d\in\mathrm{Dep}}y_{kd}=1,\ \forall k\in K; \tag{5}$$

$$\sum_{d\in\mathrm{Dep}}\sum_{k\in K}y_{kd}\leqslant L; \tag{6}$$

$$x_{ijk}\in\{0,1\},\ \forall i,j\in V,\ k\in K; \tag{7}$$

$$y_{kd}\in\{0,1\},\ \forall k\in K,\ d\in\mathrm{Dep}. \tag{8}$$

式(1)为最小化总运输路径目标函数;式(2)表示所有顾客节点出入边数相等;式(3)表示所有顾客节点有且仅有唯一车辆服务;同时式(2)和(3)保证所有顾客节点出入边数均为1;式(4)表示所有车辆载重不得超过车辆最大容量;式(5)表示每辆车归属配送中心唯一;式(6)表示所有车辆总和不得超过最大车辆数;式(7)为路径决策变量;式(8)为车辆归属决策变量.

2　算法描述

首先定义MDVRP的多智能体强化学习形式;其次设计一种基于编码器-解码器结构的策略网络,通过策略梯度强化学习方法训练网络模型;最后采用不同的动作选择策略和搜索策略获得更高质量的解.

2.1　多智能体强化学习形式定义

建立MDVRP的马尔可夫决策过程(Markov decision process, MDP),定义其状态空间、动作空间、状态转移以及回报函数,形成MDVRP的多智能体强化学习形式.

状态:状态$S=\{S_g,S_a\}$分为全局状态S_g和智能体状态$S_a=\{S_{a,1},S_{a,2},\dots,S_{a,D}\}$. 全局状态$S_g$为编码器输出的整体图特征信息,属于静态状态;智能体状态S_a由所有智能体的状态组成,单个智能体d的状态为$S_{a,d}=\{\mathrm{last}_d,\mathrm{rest}_d\}$,$\mathrm{last}_d$为智能体$d$上一步选择的节点特征,$\mathrm{rest}_d$为智能体$d$当前车辆剩余容量,$S_{a,d}$属于动态状态,随时间改变.

动作:多智能体强化学习动作空间为所有智能体的联合动作空间$A^t=\{A_d^t\}$,其中$d=1,2,\dots,D$. 智能体动作A_d^t为智能体d在当前时间步t所选择的节点,包括还未访问过的顾客点和该智能体对应的配送中心点.

状态转移:经过当前时间步t,智能体d选择动作A_d^t之后,智能体状态转移为$S_{a,d}^t=\{S_{a,d}^{t-1}*A_d^t\}$,符号"$*$"表示将动作选择的节点加入当前状态,直到形成完整的联合动作A^t之后完整状态从S^{t-1}转移为S^t.

回报:对于MDVRP而言,目标函数是最小化总距离,总距离越小则智能体的累计回报越高,因此将总距离的负数作为累计回报,有

$$R=-\sum_{d=1}^{D}\sum_{t=1}^{T-1}\mathrm{Dis}(A_d^t,A_d^{t+1}). \tag{9}$$

参数化的随机策略π_θ在每个时间步t根据策略网络输出的概率向量p_θ选择动作A^t,直到结束状态(即所有的顾客点都被访问完). 最终策略输出的解是完整的节点选择序列,即$\pi = \{\pi_1, \pi_2, \dots, \pi_T\}$,其中$T$为选择节点序列长度. 根据链式法则可知,随机策略$\pi_\theta$输出实例$s$的一个完整策略$\pi$的概率为

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

2.2 策略网络模型

策略网络包含输入、编码器、多个智能体网络构成的解码器以及输出.编码器将输入映射到隐状态,并计算得到整体图特征信息;解码器在每个时间步输出每个动作的选择概率,根据概率策略性地选择具体动作并更新状态,直到结束状态.

2.2.1 编 码 器

编码器结构如图2所示,由嵌入层和N个相同结构但网络参数相互独立的注意力模块构成,每个注意力模块由一个多头注意力层(multi-head attention, MHA)和一个前馈层(feed forward, FF)构成,输入为每个节点的特征,输出为每个节点的高层特征表示以及整体图特征信息.

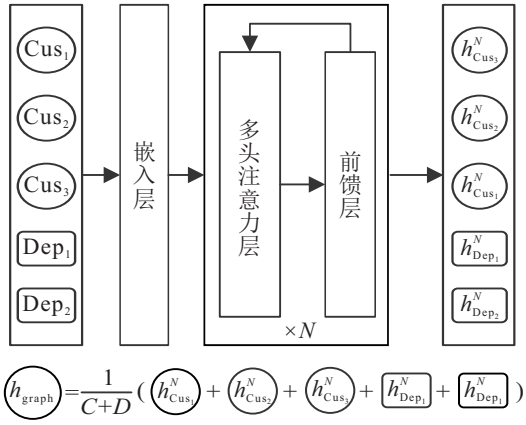


图 2 编码器网络结构

step 1: 嵌入层. 将配送中心点和顾客点的特征作为输入$X = \{x_i\}, \forall i \in V, x_i = [c_{xi}; c_{yi}; \lambda_i]$. 其中:“;”为不同特征之间的拼接,$c_x$、$c_y$为节点在平面坐标系内的横坐标和纵坐标,$\lambda_i$为节点的配送需求(配送中心点的配送需求为0). 嵌入层将每个输入x_i映射到节点嵌入特征h_i^0(特征维度$\dim(h_i^0) = 128$),有

$$h_i^0 = W^X \times x_i + b^X, \tag{11}$$

其中W^X和b^X为嵌入层网络参数.

step 2: 注意力模块. 将节点嵌入特征h^0作为注意力模块的初始输入,每一个模块包含一个MHA层和一个FF层,在每层MHA和FF均使用残差连接和批归一化处理(batch-normalization). 注意力模块将上一层节点特征h^{l-1}更新为h^l,其中上标$l \in [1, N]$表示第l个注意力模块,有

$$\hat{h}^l = \mathrm{BatchNorm}^l(h^{l-1} + \mathrm{MHA}^l(h^{l-1})), \tag{12}$$

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

step 2.1: 多头注意力层. MHA是在多个不同维度空间上的注意力机制[16],多头注意力机制有助于从不同维度获取特征信息,本文采用的注意力头数$M = 8$,即在8个维度大小为$\dim(h)/M = 16$的空间分别进行注意力计算,有

$$q_m = W_m^Q \times h^{l-1},\ k_m = W_m^K \times h^{l-1},$$
$$v_m = W_m^V \times h^{l-1}. \tag{14}$$

其中:q_m、k_m、v_m分别为在第m个维度空间上计算得到的query、key、value,W_m^Q、W_m^K、W_m^V为对应的网络参数.

与文献[15]处理VRP的方式不同,本文根据MDVRP特点重新定义节点邻居结构:如果该节点为配送中心点,则对应邻居节点为所有顾客点;如果该节点为顾客点,则对应邻居节点为所有节点. 在每一个注意力头维度空间上计算$q_{i,m}$与$k_{j,m}$之间的缩放点乘值u_{ij}^m,并利用SoftMax函数将u_{ij}^m归一化为注意力分数$a_{ij}^m \in [0, 1]$,将注意力分数a_{i*}^m与对应的$v_{i,m}$进行点积运算,得到每个注意力头子空间特征$h'_{i,m}$,最后将所有注意力头维度空间的特征融合为完整的节点特征,即

$$u_{ij}^m = \begin{cases} \dfrac{q_{i,m}^{\mathrm{T}} \times k_{j,m}}{\sqrt{\dim(k_{j,m})}},\ j \text{ adjacent to } i; \\ -\infty,\ \text{otherwise}. \end{cases} \tag{15}$$

$$a_{ij}^m = \mathrm{SoftMax}(u_{ij}^m) = \frac{\mathrm{e}^{u_{ij}^m}}{\sum\limits_{j'} \mathrm{e}^{u_{ij'}^m}}. \tag{16}$$

$$h'_{i,m} = \sum_j a_{ij}^m \times v_j^m. \tag{17}$$

$$\mathrm{MHA}(h_i) = \sum_{m=1}^{M} W_m^O \times h'_{i,m}. \tag{18}$$

其中W_m^O为多头注意力特征融合的网络参数.

step 2.2: 前馈网络层. 前馈网络层由两层线性全连接层构成,利用ReLU函数激活神经元,有

$$\mathrm{FF}(\hat{h}_i) = W_2^F \times \mathrm{ReLU}(W_1^F \times \hat{h}_i + b_1^F) + b_2^F. \tag{19}$$

其中:W_1^F、b_1^F为第1层全连接层网络参数,W_2^F、b_2^F为第2层全连接层网络参数.

2.2.2 解 码 器

解码器的多智能体框架利用上一个时间步状态信息$S^{t-1} = \{S_g, S_a^{t-1}\}$获得上下文相关向量,根据

该向量输出当前时间步动作的选择概率向量,通过选择策略选择下一步动作. 解码器具体网络结构如图3所示,其由多个智能体网络构成,每个智能体网络由嵌入层、多头注意力层和单头的注意力层组成.

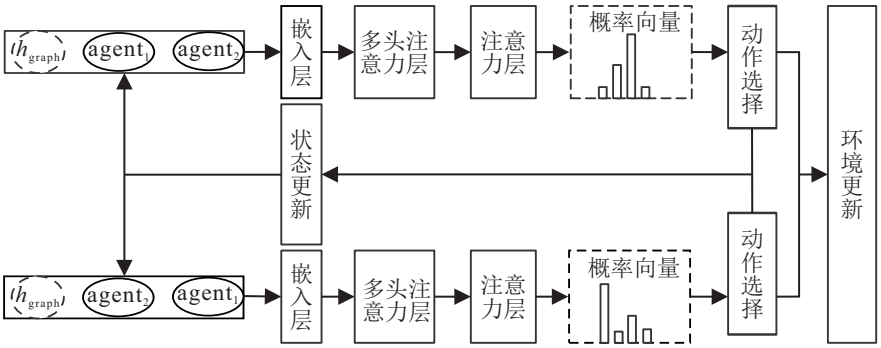


图 3　解码器网络结构

step 1: 嵌入层. 嵌入层将智能体 d 状态特征 h_d^c 映射为该智能体的上下文相关向量 $\hat{q}_d^c$,智能体 d 状态特征 h_d^c 包括整体图特征信息 h_{graph}、当前智能体特征 $\{\text{last}_d, \text{rest}_d\}$ 和其他智能体特征 $\{\text{last}_i, \text{rest}_i\}$, $i = 1, 2, \ldots, D$ 且 $i \neq d$,有

$$h_d^c = \text{concat}(h_{\text{graph}}, \text{last}_d, \text{rest}_d, \{\text{last}_i, \text{rest}_i\}), \quad (20)$$

$$\hat{q}_d^c = W^C \times h_d^c + b^C, \quad (21)$$

其中 W^C、b^C 为解码器嵌入层的网络参数.

step2: 多头注意力层. 解码器中多头注意力层的 query 由嵌入层输出的上下文相关向量 $\hat{q}_d^c$ 计算得到, key 和 value 由编码器输出的节点特征计算得到,即

$$\hat{k}_{d,m} = \hat{W}_{d,m}^K \times h^N,\ \hat{v}_{d,m} = \hat{W}_{d,m}^V \times h^N. \quad (22)$$

其中: $\hat{W}_{d,m}^K$、$\hat{W}_{d,m}^V$ 分别为计算 key 和 value 的网络参数, $m = 1, 2, \ldots, M$ 为每个注意力头维度空间, d 为当前智能体.

解码器的多头注意力层利用遮掩机制对不可访问节点进行遮掩,计算 query 与 key 之间的缩放点乘向量 $\hat{u}_{d,m}$,并利用 SoftMax 函数得到其归一化后的注意力分数 $\hat{a}_{d,m}$,将 M 头注意力空间融合获得更新的上下文相关向量 q_d^c. 其中遮掩机制描述如下: 对于每个节点 $j \in V$,若不满足如下任意一项条件,则对其遮掩: 1) j 为当前智能体对应配送中心 d 的邻居节点; 2) j 节点未被访问; 3) j 的需求小于当前智能体对应车辆剩余容. 且有

$$\hat{u}_{d,j,m} = \begin{cases} \dfrac{(\hat{q}_d^c)^{\mathrm{T}} \times \hat{k}_{d,j,m}}{\sqrt{\dim(\hat{k}_{d,j,m})}}, \text{ 满足条件1) } \sim \text{3);} \\ -\infty, \text{ otherwise.} \end{cases} \quad (23)$$

$$\hat{a}_{d,j,m} = \text{SoftMax}(\hat{u}_{d,j,m}) = \frac{\mathrm{e}^{\hat{u}_{d,j,m}}}{\sum\limits_{j'} \mathrm{e}^{\hat{u}_{d,j',m}}}. \quad (24)$$

$$\hat{q}_{d,m}^c = \sum_j \hat{a}_{d,j,m} \times \hat{v}_{d,j,m}. \quad (25)$$

$$q_d^c = \sum_{m=1}^{M} W_{d,m}^O \times \hat{q}_{d,m}^c. \quad (26)$$

其中 $W_{d,m}^O$ 为解码器多头注意力特征融合的网络参数.

step 3: 注意力层. 解码器通过一个单头的注意力层输出节点选择概率向量. 该注意力层计算 query 与 key 之间的相容度 $u_{d,j}$, query 由经过多头注意力层更新的上下文相关向量 q_d^c 计算得到, Key 由节点特征计算得到,即 $k_{d,j} = W_d^K \times h_j^N$. 利用 SoftMax 函数进行归一化,得到每个节点的选择概率 $p_{d,j}$,有

$$u_{d,j} = \begin{cases} C \times \tanh\left(\dfrac{(q_d^c)^{\mathrm{T}} \times k_{d,j}}{\sqrt{\dim(k_{d,j})}}\right), \text{ 满足条件1) } \sim \text{3);} \\ -\infty, \text{ otherwise.} \end{cases} \quad (27)$$

$$p_{d,j} = \text{SoftMax}(u_{d,j}) = \frac{\mathrm{e}^{u_{d,j}}}{\sum\limits_{j'} \mathrm{e}^{u_{d,j'}}}. \quad (28)$$

其中 $\tanh(x) = \dfrac{\mathrm{e}^x - \mathrm{e}^{-x}}{\mathrm{e}^x + \mathrm{e}^{-x}}$ 为双曲正切激活函数,并将结果缩放到 $[-C, C]$,本文设置 $C = 10$.

根据每个智能体输出的概率向量 p_d,由动作选择策略选择该智能体的下一步动作 A_d^t,得到联合动作 $A^t = \{A_1^t, A_2^t, \ldots, A_D^t\}$,直到所有顾客访问完,形成完整的策略解 $\pi = \{\pi_1, \pi_2, \ldots, \pi_T\}$.

2.3　策略网络训练方法

带有回滚基准的 REINFORCE 算法[24] 主要基于策略的强化学习算法,该算法通过计算单智能体的累计回报估计策略梯度并训练单智能体策略. 本文采用该算法训练多智能体联合策略,通过计算联合

策略的累计回报估计策略梯度并训练多智能体策略网络. 对于一个给定的实例 s, 策略网络 θ 输出所有智能体每一步的动作概率向量 $p_\theta(\pi_t|s)$, 进而以采样选择的方式输出联合策略 $\pi_t = \text{sample}(p_\theta(\pi|s))$. 而基准网络 θ^{bl} 则根据基准网络输出的动作概率向量 $p_{\theta^{bl}}(\pi_t|s)$ 以贪婪选择的方式输出联合策略 $\pi_t^{bl} = \text{greedy}(p_{\theta^{bl}}(\pi|s))$. 根据蒙特卡洛算法评估策略的期望累计回报 $L(\theta|s) = E_{p_\theta(s)}[R(\pi)]$, 其中 $R(\pi)$ 为策略 $\pi = \{\pi_1, \pi_2, \ldots, \pi_T\}$ 的累计回报. 利用带有基准的REINFORCE算法计算策略梯度, 并采取梯度下降的方式更新策略网络参数, 即

$$\nabla_\theta L(\theta|s) = -E_{p_\theta(\pi|s)}[(R(\pi) - R(\pi^{bl})) \times \nabla_\theta \log p_\theta(\pi|s)], \quad (29)$$

$$\theta = \text{Adam}(\theta, \nabla_\theta L(\theta|s)). \quad (30)$$

利用基准网络 θ^{bl} 评估实例 s 的难易程度, 可以有效降低训练网络时梯度的方差. 基准网络以回滚的方式更新, 在每一轮训练结束对策略网络 θ 与基准网络 θ^{bl} 进行比较, 在显著性水平为 $\alpha(=0.05)$ 的 t 检验中, 如果策略网络输出的解显著优于基准网络, 则对基准网络进行回滚式更新 $\theta^{bl} \to \theta$.

2.4 动作选择及局部搜索策略

经过多轮次学习的策略网络拥有较好的决策能力, 动作选择策略根据网络输出的概率向量选择动作. 本文采用两种不同的动作选择策略:

1) 贪婪动作选择策略完全信任策略网络, 每一步都以解码器输出概率值为基准, 选择拥有最大概率值的动作;

2) 采样动作选择策略以解码器输出概率作为采样概率分布, 在该分布上进行动作采样选择, 因此该策略并非每次都会选择最大概率值的动作, 而是以不同的概率选择对应动作.

在网络训练过程中基准网络 θ^{bl} 作为每一批次训练实例难易程度的评判标准充当了“评价者”的角色, 采用贪婪动作选择策略可以快速获得有效的“评价指标”. 而策略网络 θ 作为“执行者”, 需要对其决策能力进行有效评估, 利用采样动作选择策略可以有效估计解质量的期望值, 即该“执行者”的决策能力. θ^{bl} 和 θ 通过不同的动作选择策略选择合适的动作能够有效提高学习效率和模型性能.

训练完的MADRL-model采用贪婪动作选择策略可以快速求解MDVRP, 但针对部分较难的实例存在一定的改进空间, 例如子回路中路线交叉问题以及贪婪动作选择策略的过度自信行为导致错失了拥有较高选择概率(但并非最高)的动作. 为了改进解的质量, 本文采用两种局部搜索策略:

1) 2-opt搜索. 针对出现的子回路中路线交叉问题, 2-opt局部搜索将模型输出的解的每个子回路作为初始解, 对所有子回路进行2-opt操作进行寻优, 从而改进解的整体质量, 具体步骤如下.

step 1: 随机选择当前子回路 r 的两个节点, 并对该节点对之间的路径进行翻转, 形成新的子回路 r';

step 2: 若子回路 r' 优于 r, 则更新当前子回路 $r = r'$, 将迭代次数iter重置为0并返回step 1, 否则迭代次数 $\text{iter} = \text{iter} + 1$ 并返回step 1;

step 3: 若迭代次数达到最大迭代次数 $\text{Iter} = \text{MaxIter}$ 仍未改进 r, 则结束2-opt局部搜索, 并将 r 作为最优子回路返回.

2) 采样搜索. 模型使用采样动作选择策略重复求解同一个实例以获得该实例多个完整解, 取其中最优的解, 即设 s 为完整解采样个数, 则最优解 $\pi* = \text{argmin}\{L(\pi_1), L(\pi_2), \ldots, L(\pi_s)\}$, 避免了策略网络的过度自信行为. 由于本文模型具有快速求解的优势, 重复采样求解并不会消耗十分昂贵的时间成本.

3 实验分析

3.1 实验设置

使用pytorch实现MADRL-model整体框架, 在单张GPU (2080ti, 显存为10 G) 上训练策略网络模型, 在运行环境为Intel Core i7 CPU/3.60 GHz的win10操作系统上进行算例测试. 为验证MADRL-model性能, 分别在20-3规模(20个顾客点, 3个配送中心点, 车辆最大容量为30)、50-3规模(50个顾客点, 3个配送中心点, 车辆最大容量为40)和100-2规模(100个顾客点, 2个配送中心点, 车辆最大容量为50)的算例上训练模型, 算例生成方式参考文献[15]中CVRP实例的生成方式, 节点的坐标在 $[0,1] \times [0,1]$ 上均匀分布, 顾客点需求为 $[1,10]$ 上的均匀分布.

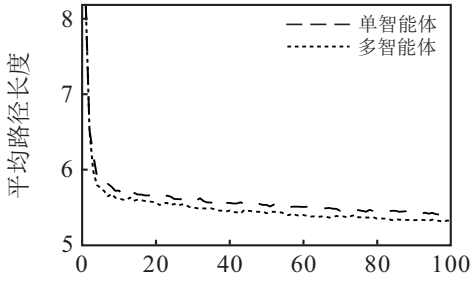
3.2 参数设置

在模型训练环节, 对于20-3规模和50-3规模问题, 训练轮数(epoch)设置为100, 每轮训练批数设置为2 500, 每批次算例数设置为512; 对于100-2规模问题, 由于显存大小限制, 每批次算例数设置为128, 采样搜索中的采样数 s 设置为128. 采用Adam优化器[25]优化策略网络参数, 学习率设置为 1×10^{-4}. 在算例测试环节, 对于不同规模的问题, 分别在其对应分布下测试10 000组算例, 将所有测试算例路径长度平均值和算法平均求解时间作为模型性能评判指标, 平均路径长度越短表示策略越优, 平均求解时间越短

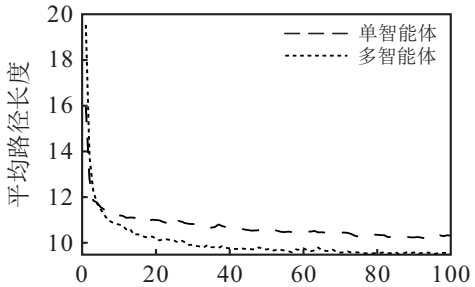
表示模型效率越高.

3.3　多智能体框架的有效性检验

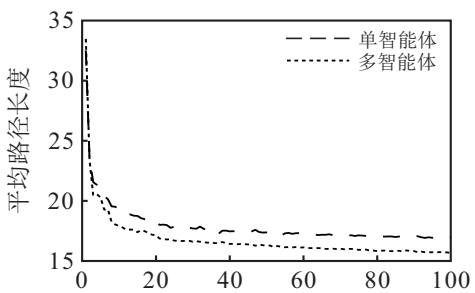
在单智能体强化学习中,策略网络含有单个智能体结构,只需要智能体与环境之间的交互,但是MADRL-model中含有多个智能体,不仅需要对环境有准确认知,还需要将不同智能体之间的特征信息进行融合达到相互合作的效果. 多智能体之间的交互通过网络模型中解码器嵌入层的特征融合实现,解码器将当前智能体状态特征与其他智能体状态特征融合,使得当前智能体选择节点时考虑联合动作最优;而单智能体学习模式中解码器嵌入层只包含当前智能体特征,因此只考虑当前智能体动作最优. 本文设置了多智能体框架的有效性检验实验,将MADRL-model训练结果与单智能体学习模式进行对比,以验证多智能体框架的有效性. 由图4分析可知,在20-3规模问题上,MADRL-model最终训练结果略优,而在50-3规模和100-2规模问题上,MADRL-model训练结果明显占优,且相比单智能体学习模式,MADRL-model在训练过程中收敛速度更快,收敛效果好于单智能体学习模式.



(a)　20-3规模



(b)　50-3规模

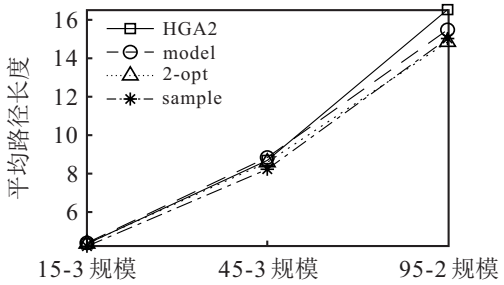


(c)　100-2规模

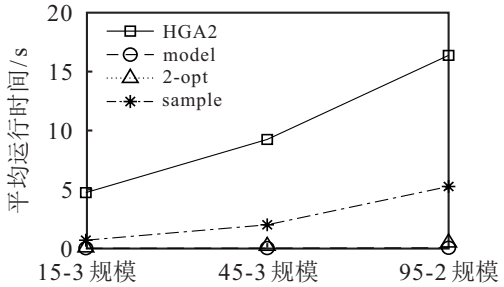
图4　多智能体与单智能体训练过程对比

3.4　模型迁移性能检验

不同规模的MADRL-model模型具有一定的迁移能力,例如对于50-3规模问题,训练模型所用算例包含50个顾客点,但是该模型在求解包含45个顾客点的算例时同样有效. 为验证MADRL-model模型的迁移性能,将不同规模模型进行迁移求解并与HGA2[25]进行对比,20-3规模MADRL-model模型求解15-3规模算例,50-3规模MADRL-model模型求解45-3规模算例,100-2规模MADRL-model模型求解95-2规模算例. 对比结果如图5所示. 在求解结果和求解时间上,MADRL-model以及结合搜索策略在不同问题规模上的表现均优于HGA2,并且随着规模增大,模型的优势更加明显,表明本文模型具有良好的迁移能力.



(a)　平均路径长度



(b)　平均求解时间

图5　模型迁移求解对比

3.5　与其他算法的性能对比

将MADRL-model以及结合2-opt局部搜索策略和采样搜索策略与求解MDVRP的常用算法HGA[25]、GVNS[7]进行对比. HGA是一种结合邻域搜索的遗传算法,其中HGA1采用随机初始化生成初始种群,HGA2采用节约算法[6]和NNH生成初始种群,并在遗传操作中增加ISP算子;GVNS是一种变邻域搜索算法,该算法采用Gillett等[1]提出的扫描算法生成初始解,并采用改进的随机变邻域下降算法(randomized variable neighborhood descent, RVND)作为局部搜索方法更新最优解. 对比算法中的参数设置均与对应文献一致,HGA中种群规模为25,交叉概率为0.4,变异概率为0.2,对于20-3规模和50-3规模问题迭代次数为500,对于100-2规模问

题迭代次数为1 000; GVNS中最大迭代次数为100, 最大迭代时间为30 min. 结果如表1和表2所示, 其中MADRL-model为所提出模型经过离线训练的求解结果, Model(2-opt)表示对MADRL-model求得的解采用2-opt局部搜索策略, Model(sample)表示对MADRL-model求得的解采用采样搜索策略.

由表1表2可见: MADRL-model在20-3规模问题和50-3规模问题上的求解质量与对比算法相近, 但是在求解时间上远远快于其他算法, 而在100-2规模问题上不管是求解质量还是求解时间均优于其他算法; 带有2-opt局部搜索策略和采样策略的MADRL-model在3种规模问题上的求解质量和求解时间均优于其他算法. 通过对3种不同策略的MADRL-model对比分析可知, 2-opt局部搜索策略和采样搜索策略在解的质量上均优于模型求解结果, 由于增加了不同的搜索策略, 其求解时间关系近似于 $\mathrm{Time}_{\mathrm{sample}} \approx 10\,\mathrm{Time}_{\text{2-opt}} \approx 100\,\mathrm{Time}_{\mathrm{model}}$. 由于策略网络输出为每一步动作的选择概率, 对于不同实例问题策略网络可能会出现最优动作的概率并非最高的情况, 采样策略有一定概率在较高(非最高)概率的动作中选择, 由采样策略优于局部搜索策略改进MADRL-model解的结果可知, 最优动作虽然不一定概率最高但是极有可能是较高概率动作, 通过采样策略即可大概率获得最优动作.

表1 不同算法求解结果对比

算 法	问题规模		
	20-3	50-3	100-2
HGA1	5.726 6	11.487 3	20.485 3
HGA2	5.257 4	9.490 0	16.336 1
GVNS	5.427 9	10.362 2	18.857 3
MADRL-model	5.287 8	9.477 2	15.619 0
model(2-opt)	5.220 0	9.231 5	15.124 9
model(sample)	5.073 3	8.861 1	15.071 7

表2 不同算法求解时间对比

算 法	问题规模		
	20-3	50-3	100-2
HGA1	5.364 3	10.059 9	17.075 9
HGA2	5.550 8	10.421 9	17.647 9
GVNS	1.509 1	15.591 6	37.224 3
MADRL-model	0.007 8	0.027 3	0.043 9
model(2-opt)	0.067 5	0.208 2	0.483 2
model(sample)	0.614 0	2.091 4	5.680 4

4 结 论

本文提出了一种基于多智能体深度强化学习框架的MDVRP求解模型, 区别于传统“先分组后规划”的求解思路, MADRL-model的多智能体利用高层特征信息, 通过学习相互合作的动作, 从问题整体进行车辆路径规划, 经过离线训练的MADRL-model模型可以快速求解MDVRP. 通过算例仿真对比实验验证了MADRL-model模型与启发式算法相比具有更快的求解速度, 并且从整体进行求解, 使得配送中心之间、顾客点分布之间具有更好的完整性, 解的质量更优. 后续研究将进一步考虑模型在配送中心数量变动情况下的泛化能力以及更大规模问题的求解能力, 并设计更有效的求解模型.

参考文献(References)

[1] Gillett B E, Johnson J G. Multi-terminal vehicle-dispatch algorithm[J]. Omega, 1976, 4(6): 711-718.

[2] Bettinelli A, Ceselli A, Righini G. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows[J]. Transportation Research — Part C: Emerging Technologies, 2011, 19(5): 723-740.

[3] He Y L, Miao W D, Xie R, et al. A tabu search algorithm with variable cluster grouping for multi-depot vehicle routing problem[C]. Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design. Hsinchu, 2014: 12-17.

[4] Oliveira F B, Enayatifar R, Sadaei H J, et al. A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem[J]. Expert Systems with Applications, 2016, 43: 117-130.

[5] Ho W, Ho G T S, Ji P, et al. A hybrid genetic algorithm for the multi-depot vehicle routing problem[J]. Engineering Applications of Artificial Intelligence, 2008, 21(4): 548-557.

[6] Clarke G, Wright J W. Scheduling of vehicles from a central depot to a number of delivery points[J]. Operations Research, 1964, 12(4): 568-581.

[7] Bezerra S N, de Souza S R, Souza M J F. A GVNS algorithm for solving the multi-depot vehicle routing problem[J]. Electronic Notes in Discrete Mathematics, 2018, 66: 167-174.

[8] 王征, 张俊, 王旭坪. 多车场带时间窗车辆路径问题的变邻域搜索算法[J]. 中国管理科学, 2011, 19(2): 99-109.
(Wang Z, Zhang J, Wang X P. A modified variable neighborhood search algorithm for the multi depot vehicle routing problem with time windows[J]. Chinese Journal of Management Science, 2011, 19(2): 99-109.)

[9] 许维胜, 曾正洋, 徐志宇. 一种求解两级车辆路径问题的Memetic算法[J]. 控制与决策, 2013, 28(10): 1587-1590.
(Xu W S, Zeng Z Y, Xu Z Y. A Memetic algorithm for solving two-echelon vehicle routing problem[J]. Control and Decision, 2013, 28(10): 1587-1590.)

[10] 曾正洋, 许维胜, 徐志宇, 等. 应急物流中的累计时间式多车场车辆路径问题[J]. 控制与决策, 2014, 29(12): 2183-2188.
(Zeng Z Y, Xu W S, Xu Z Y, et al. Cumulative multi-depot vehicle routing problem in emergency logistics[J]. Control and Decision, 2014, 29(12): 2183-2188.)

[11] 周鲜成, 吕阳, 贺彩虹, 等. 考虑时变速度的多车场绿色车辆路径模型及优化算法[J]. 控制与决策, 2022, 37(2): 473-482.
(Zhou X C, Lv Y, He C H, et al. Multi-depot green vehicle routing model and its optimization algorithm with time-varying speed[J]. Control and Decision, 2022, 37(2): 473-482.)

[12] Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]. The 29th Conference on Neural Information Processing Systems. Montreal, 2015: 2692-2700.

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

[14] Nazari M, Oroojlooy A, Takac M, et al . Reinforcement learning for solving the vehicle routing problem[C]. The 32nd Conference on Neural Information Processing Systems. Montreal, 2018: 9861-9871.

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

[16] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. The 31st Advances in Neural Information Processing Systems. Los Angeles, 2017: 5998-6008.

[17] Li Z, Chen Q, Koltun V . Combinatorial optimization with graph convolutional networks and guided tree search[C]. The 32nd Conference on Neural Information Processing Systems. Montreal, 2018: 537-546.

[18] Nowak A, Villar S, Bandeira AS, et al. A note on learning algorithms for quadratic assignment with graph neural networks[C]. The 34th International Conference on Machine Learning. Sydney, 2017: 1-12.

[19] Helsgaun K. An extension of the Lin-kernighan-helsgaun TSP solver for constrained traveling salesman and vehicle routing problems[R]. Roskilde: Roskilde University, 2017.

[20] Chen X Y, Tian Y D. Learning to perform local rewriting for combinatorial optimization[J/OL]. 2018, arXiv: 1810.00337.

[21] Lu H, Zhang X W, Yang S. A learning-based iterative method for solving vehicle routing problems[C]. The 8th International Conference on Learning Representations. Addis Ababa, 2020: 1-12.

[22] Costa P, Rhuggenaath J, Zhang Y Q, et al. Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning[C]. The 12th Asian Conference on Machine Learning. Bangkok, 2020: 465-480.

[23] Wu Y, Song W, Cao Z, et al. Learning improvement heuristics for solving routing problems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 8(1): 1-13.

[24] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3/4): 229-256.

[25] Kingma D P, Ba J L. Adam: A method for stochastic optimization[C]. The 3rd International Conference on Learning Representations. San Diego, 2015: 1-11.

作者简介

王万良 (1957−), 男, 教授, 博士生导师, 从事人工智能、大数据等研究, E-mail: zjutwwl@zjut.edu.cn;

陈浩立 (1995−), 男, 硕士生, 从事智能优化调度的研究, E-mail: chenhaoli1222@163.com;

李国庆 (1994−), 男, 博士生, 从事多目标优化的研究, E-mail: zjutwwl@zjut.edu.cn;

冷龙龙 (1991−), 男, 博士生, 从事智能配送与优化调度的研究, E-mail: cyxlll@zjut.edu.cn;

赵燕伟 (1959−), 女, 教授, 博士生导师, 从事数字化产品现代设计理论与方法、现代物流系统智能配送与优化调度等研究, E-mail: ywz@zjut.edu.cn.

(责任编辑：郑晓蕾)