中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

基于多起点和Mask策略的深度强化学习算法求解覆盖旅行商问题

方伟, 接中冰, 陆恒杨, 张涛

引用本文:
方伟,接中冰,陆恒杨,张涛. 基于多起点和Mask策略的深度强化学习算法求解覆盖旅行商问题[J]. 控制与决策, 2024, 39(4): 1160−1166.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2022.0755

基于多起点和Mask策略的深度强化学习算法求解覆盖旅行商问题

方 伟[1,2], 接中冰[1,2], 陆恒杨[1,2†], 张 涛[3]

(1. 江南大学 江苏省人工智能国际合作联合实验室, 江苏 无锡 214122；2. 江南大学 江苏省模式识别与计算智能工程实验室，江苏 无锡 214122；3. 中国船舶科学研究中心，江苏 无锡 214082)

摘 要: 覆盖旅行商问题(covering salesman problem, CSP) 是旅行商问题的变体,在防灾规划、急救管理中有着广泛应用. 由于传统方法求解问题实例耗时严重,近年来深度神经网络被提出用于解决该类组合优化问题,在求解速度和泛化性上有明显的优势. 现有基于深度神经网络求解CSP的方法求解质量较低,特别在大规模实例上与传统的启发式方法相比存在较大差距. 针对上述问题,提出一种新的基于深度强化学习求解CSP的方法,由编码器对输入特征进行编码,提出新的Mask策略对解码器使用自注意力机制构造解的过程进行约束,并提出多起点策略改善训练过程、提高求解质量. 实验结果表明,所提方法对比现有基于深度神经网络的求解方法进一步缩小了最优间隙,同时有着更高的样本效率,在不同规模和不同覆盖类型的CSP中展现出更强的泛化能力,与启发式算法相比在求解速度上有10～40倍的提升.

关键词: 覆盖旅行商；深度强化学习；组合优化；多起点；Mask策略

中图分类号: TP399 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2022.0755

引用格式: 方伟,接中冰,陆恒杨,等. 基于多起点和Mask策略的深度强化学习算法求解覆盖旅行商问题[J]. 控制与决策, 2024, 39(4): 1160-1166.

Deep reinforcement learning algorithm based on multi-start and Mask strategy for solving the covering salesman problem

FANG Wei[1,2], *JIE Zhong-bing*[1,2], *LU Heng-yang*[1,2†], *ZHANG Tao*[3]

(1. International Joint Laboratory on Artificial Intelligence of Jiangsu Province，Jiangnan University，Wuxi 214122，China；2. Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence，Jiangnan University，Wuxi 214122，China；3. China Ship Scientific Research Center，Wuxi 214082，China)

Abstract: The covering salesman problem (CSP) is a variant of the traveling salesman problem, which is widely used in disaster planning and emergency management. Since the traditional solvers are time-consuming for solving problem instances, deep neural networks have been proposed for solving this type of combinatorial optimization problem in recent years, which have obvious advantages in terms of solving speed and generalization. However, existing methods based on deep neural networks for solving CSP problems have low solution quality, especially in large-scale instances, compared with traditional heuristics. Therefore, we propose a new method based on deep reinforcement learning to solve the CSP problem, in which the encoder encodes the input features, propose a new Mask strategy to constrain the decoder to construct a solution using the self-attention mechanism, and propose a multi-start strategy to improve the training process and the solution quality. Experimental results show that the proposed method further reduces the optimality gap compared with existing deep neural network-based solution methods, has higher sample efficiency, shows stronger generalization ability in CSP tasks of different sizes and coverage types, and has a 10-40 times improvement in solution speed compared with heuristic algorithms.

Keywords: covering salesman problem；deep reinforcement learning；combinatorial optimization；multi-start；Mask strategy

收稿日期: 2022-05-04；**录用日期:** 2022-12-01.

基金项目: 国家自然科学基金项目(62073155, 62002137, 62106088, 62206113)；船舶总体性能创新研究开放基金项目(22422213).

责任编委: 李少远.

†通讯作者. E-mail: luhengyang@jiangnan.edu.cn.

*本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

0 引　言

旅行商问题(traveling salesman problem, TSP)[1]是经典的组合优化问题,其求解目标是在给定的顶点集合中寻找到一条最短回路,并且所有的顶点能且只能被访问一次. 由于资源等因素的限制,该问题模型不能直接地应用到某些现实生活场景中,例如农村医疗队路线规划问题[2],医疗队访问所有的村庄并对其居民提供医疗服务需要花费较大的代价,一个更好的解决方案是未被医疗队访问的村庄村民可以去距离其最近的被访问村庄获取医疗服务. 为了研究该类问题,文献[2]设计了覆盖旅行商问题(covering salesman problem, CSP)模型. CSP模型中每个顶点存在一个预先设定的覆盖距离,处于覆盖距离内的其他顶点可被该顶点所覆盖. CSP求解目标是在顶点集合中寻找到一个子集进行排列,构成一条使所有顶点被访问或被覆盖的最短回路. 当各个顶点预先设定的覆盖距离均为0时,CSP退化为TSP. 因此,TSP可以看作CSP的特例,CSP同样具有NP难特征.

对CSP进行求解主要使用启发式算法. 文献[2]提出两阶段启发式,首先选取能够覆盖所有顶点的最小数量顶点集合,然后在该集合内进行TSP的求解. 文献[3]提出LS 1、LS 2算法,在完整解的基础上执行删除、添加操作改善解的质量. 文献[4]通过删除和插入顶点对初始解进行改进,并结合整数线性规划最小化路径长度. 此外,一些群体智能优化算法[5-7]也被用于CSP的求解. 上述文献所提方法在性能方面与LS 1和LS 2算法相差不大[8],然而这些启发式求解方法需要大量的领域知识以及不断试错,难以在可行的时间内求解大规模问题实例. 同时,该类方法没有考虑实例间的内在相似性,无法获取已经求解过的问题所提供的知识.

近年来,部分学者尝试使用深度神经网络(deep neural network, DNN)求解组合优化问题[9-10],在推理时间和泛化性上展现出优势[11-13]. 文献[8]在注意力模型(attention model, AM)[10]的基础上,根据CSP特征设计了动态嵌入,提出AM-dynamic模型求解CSP,并通过简单的局部搜索算法对解质量进行改进,求解速度有大幅提升. 然而,所提出的动态嵌入并不能有效地捕捉到CSP的问题特征,特别是在顶点数量超过100的实例上性能急剧下降. 同时,由于采用模型预测单一起点构造解,求解质量与启发式算法仍存在较大差距. 鉴于此,本文提出一个新Mask策略的注意力模型(attention model with new mask, AM-NM)以求解CSP. 模型使用编码器-解码器结构,编码器采用多头自注意力机制[14]对输入的二维顶点坐标进行编码,解码器针对CSP设计新的Mask策略对解的构造进行约束,并在训练过程中通过多起点构造解提高模型的求解质量. AM-NM使用强化学习进行训练,不依赖于高质量的标注数据,在求解质量和求解速度上均取得大幅提升. 总体而言本文贡献如下:

1) 提出新的Mask策略用于AM模型求解CSP,能够有效提高模型的收敛速度和求解质量,显著优于现有DNN求解CSP的方法.

2) 使用多起点策略提高样本效率,并改善求解质量和收敛过程.

3) 所提出方法可以缩小与启发式算法的最优间隙(optimality gap),在求解速度上有超过10倍的提升,并展现出更强的泛化能力.

1 问题定义及基于深度强化学习的CSP求解方法

1.1 问题定义

定义$G=(N,\text{Edge})$为一个二维空间内的无向完全图. 其中:$N=\{1,2,\dots,n\}$为含有n个顶点的集合,$\text{Edge}=\{(i,j)|i,j\in N\}$为边集合. 定义$E$为距离矩阵,$E_{i,j}(1\leqslant i,j\leqslant n)$为边$(i,j)$的长度,即顶点$i$与顶点$j$间的欧氏距离. 每个顶点$i$存在一个预先设定的可覆盖集合$\text{Set}_i$,集合内的顶点均可被顶点$i$覆盖. CSP的求解目标是在顶点集合$N$中寻找一个排列的子序列$\pi=(\pi_1,\pi_2,\dots,\pi_k)(k\leqslant n)$,使$N$中所有顶点都被访问或被覆盖,且$\pi$构成的回路路径最短,即$\text{len}(\pi)$最小,有

$$\begin{aligned}&\min\ \text{len}(\pi)=\\&E_{\pi_k,\pi_1}+\sum_{i=1}^{k-1}E_{\pi_i,\pi_{i+1}}=\\&\|x_{\pi_1}-x_{\pi_k}\|_2+\sum_{u=1}^{k-1}\|x_{\pi_u}-x_{\pi_{u+1}}\|_2,\end{aligned}\tag{1}$$

其中x_{π_i}为顶点π_i的二维平面坐标. 图1给出了顶点集合$N=\{1,2,\dots,8\}$的CSP实例及可行解序列$\pi=(1,6,7)$. 该实例中每个顶点能够对其最近的2个顶点进行覆盖,绿色点表示解序列中的顶点,红色圆

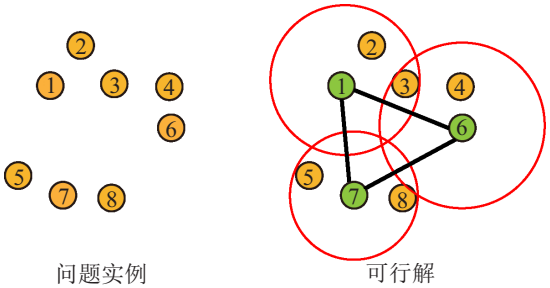


图1　CSP问题实例及可行解

圈表示解序列顶点的覆盖范围,黑色线段表示解序列构成的路径.

1.2 基于深度强化学习的CSP求解

给定CSP问题实例s,将其建模为马尔可夫决策过程. 实例s的顶点坐标和已经访问过的顶点为状态,当前时刻t选择的顶点π_t为动作. 经过k次动作后,满足解的可行性条件的动作序列π为CSP的解. 将动作序列π构成的负路径长度$-\text{len}(\pi)$定义为奖励,最大化奖励等价于最小化路径长度. 定义策略为状态到动作的映射,通常为神经网络近似的随机策略$p_\theta(\pi|s)$,该策略通过参数为θ的DNN进行参数化,可因式分解为

$$p_\theta(\pi|s)=p_\theta(\pi_1|s)\dots p_\theta(\pi_k|\pi_{k-1},\dots,\pi_1,s)=\prod_{t=1}^{k}p_\theta(\pi_t|\pi_{1:t-1},s),\ k\leqslant n. \tag{2}$$

当策略$p_\theta(\pi|s)$采样一个回合的数据并得到奖励后,根据Reinforce算法进行策略梯度估计以调整参数θ. 基于深度强化学习求解CSP的整体框架如图2所示.

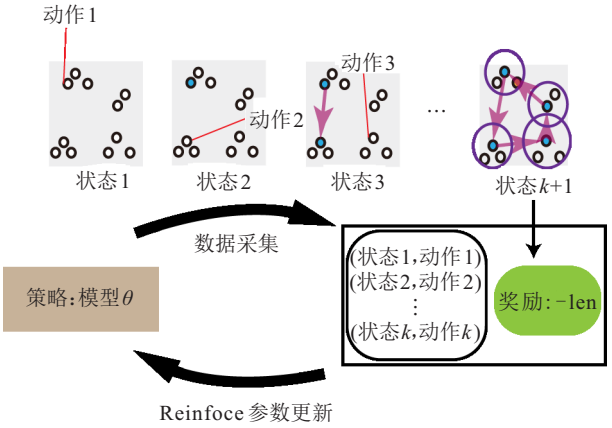


图 2 基于深度强化学习求解CSP的整体框架

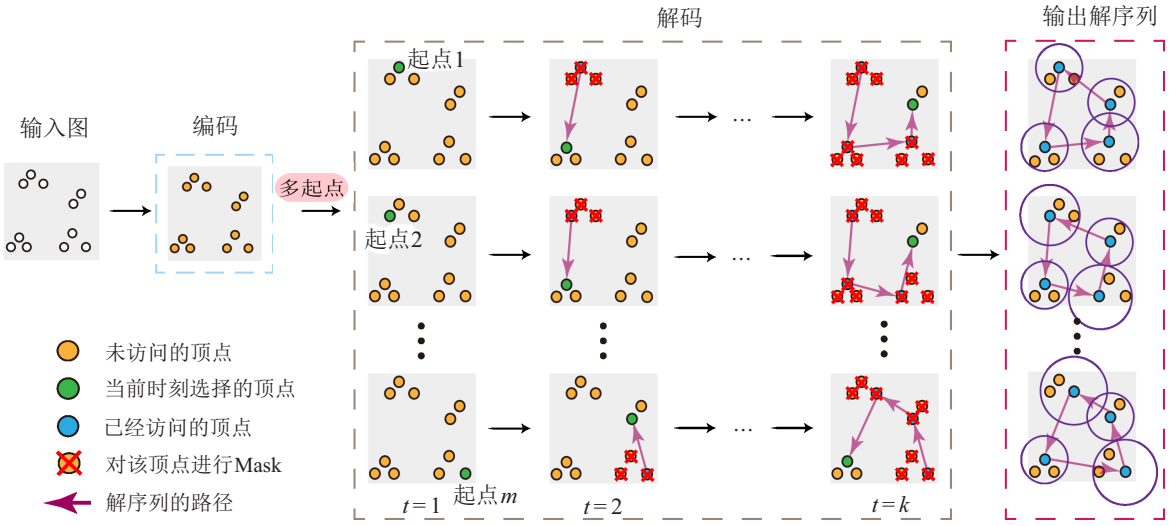


图 3 模型求解示意图

2 AM-NM: 新Mask策略的注意力模型

2.1 AM-NM

AM-NM采用编码器-解码器结构,对给定的CSP实例进行求解,过程如图3所示. 首先由编码器对输入的二维坐标进行特征编码得到顶点的嵌入(embedding);在此基础上执行多起点策略,选取m个起点通过解码器进行完整解的构造. 解码器采取自回归的方式在t时刻$(1\leqslant t\leqslant k)$从未被访问过的顶点集合中选择一个顶点进行访问,并使用Mask策略减少待预测顶点数量,直至所有顶点被访问或覆盖. 算法1给出了获取模型最优参数θ^*的训练过程.

算法1 模型训练过程.

输入: S为CSP实例训练集,Total为每轮训练的CSP实例数量,Epochs为训练轮数,bs为批次大小,m为多起点数量,θ为模型参数;

输出: 最优模型参数θ^*.

1) 初始化模型参数θ.

2) for epoch $=1,2,\dots,$Epochs do

3) for step $=1,2,\dots,\left\lceil\dfrac{\text{Total}}{\text{bs}}\right\rceil$ do

4) $S_{\text{bs}}\leftarrow$从训练集S中采样bs个实例

5) for $S_r\in S_{\text{bs}}$ do

6) $H_r\leftarrow\text{Encoder}_\theta(S_r)$

7) $\{a_1,a_2,\dots,a_m\}\leftarrow$为实例$S_r$选取$m$个起点

8) $P_r^z\leftarrow\text{Decoder}_\theta(a_z,H_r),\forall z\in\{1,2,\dots,m\}$

9) $\Pi_r^z\leftarrow$根据概率分布P_r^z随机采样获取解序列,$\forall z\in\{1,2,\dots,m\}$

10)　　end for

11)　　$\theta^* \leftarrow \text{reinforce}(\theta, P, \Pi)$

12)　end for

13) end for

本文使用标准Transformer模型编码器对输入的二维顶点坐标进行特征编码. 针对CSP具有的输入顺序无关性删除位置编码, 并使用批次归一化(batch norm, BN)替换层归一化(layer norm). 图4所示为编码器的具体结构.

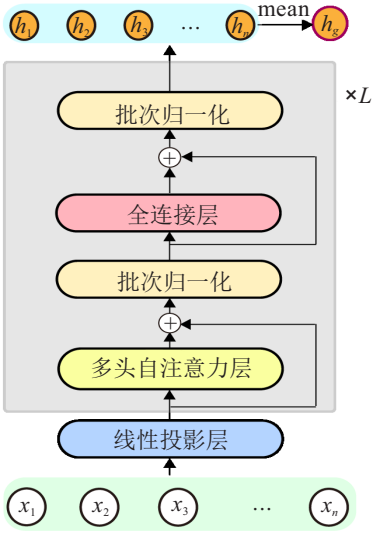


图 4　编码器结构

首先通过线性投影层将顶点i的特征x_i投射到高维空间, 即

$$h_i^0 = Wx_i + b,\ i = 1, 2, \ldots, n. \tag{3}$$

其中: $W \in R^{d\times d_x}$、$b \in R^{d\times 1}$为待学习的参数; d为隐层维度; d_x为特征x_i的维度, 通常为2; $h_i^0 \in R^d$为顶点i的初始嵌入; n个顶点的初始嵌入可通过矩阵形式表示, 即$H^0 = (h_1^0, h_2^0, \ldots, h_n^0)^{\mathrm{T}} \in R^{n\times d}$. 然后将$H^0$通过编码层进行特征提取, 其中编码层的层数为$L$, 上标$l = \{1, 2, \ldots, L\}$表示在第$l$层编码层中的运算结果. 编码层由多头自注意力(multi-head self-attention, MHA)层[14]进行顶点信息融合, 在MHA层后通过全连接(feed-forward layer, FF)层进行特征提取, 并引入残差连接和批次归一化确保深层网络训练的稳定性, 有

$$Q^l = W_q^l H^{l-1},\ K^l = W_k^l H^{l-1},\ V^l = W_v^l H^{l-1}; \tag{4}$$

$$H'^l = BN^l(MHA^l(Q^l, K^l, V^l) + H^{l-1}); \tag{5}$$

$$H^l = BN^l(H'^l + FF^l(H'^l)). \tag{6}$$

经过L层编码后, n个顶点的嵌入以矩阵形式表示为$H^L = (h_1^L, h_2^L, \ldots, h_n^L)^{\mathrm{T}}$, 并根据下式得到表示全局图信息的图嵌入$h_g$:

$$h_g = \frac{1}{n}\sum_{i=1}^{n} h_i^L. \tag{7}$$

2.2　解 码 器

根据编码后的顶点嵌入和图嵌入h_g, 每个时刻t, 利用解码器从未访问的顶点集合进行选择. 图5展示了解码器构造解的过程. 在$t = 1$时刻选择起点π_1, 在$t \geqslant 1$时刻基于已构造的部分解路径的首、尾顶点信息对$t+1$时刻访问的顶点作出预测, 并加入到解序列π中, 直至解满足可行性要求.

2.2.1　多起点的选取

在$t = 1$时刻, 解码器选择起点. 考虑到模型构造最短路径的求解能力不会因起点的不同产生差异, 而采样更多的路径能够有效降低策略梯度的方差, 帮助模型更快地收敛, 因此对单个实例s选取多个起点进

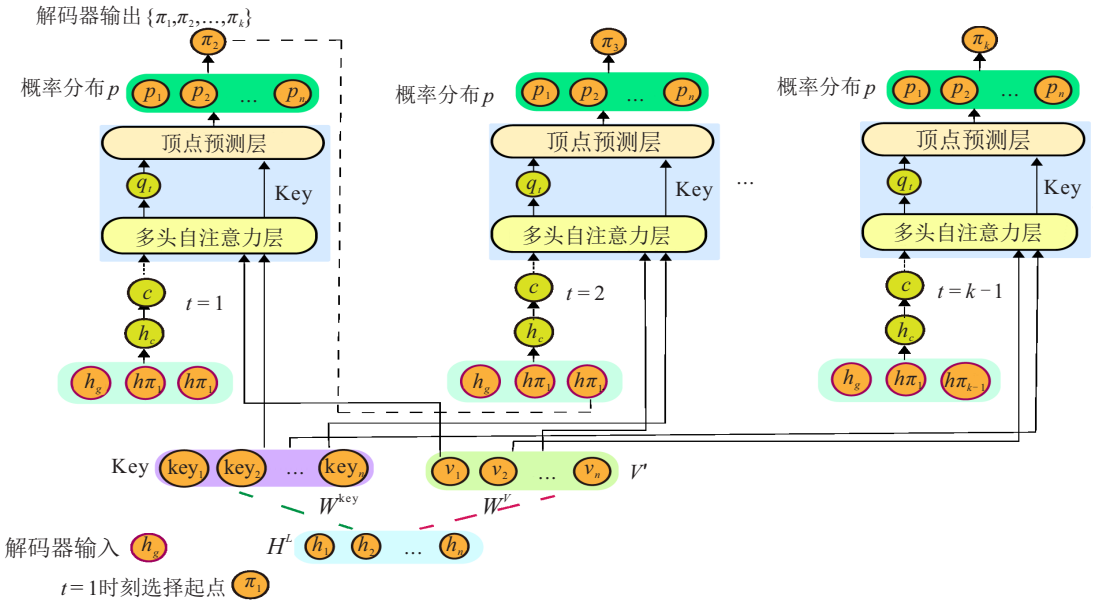


图 5　解码器结构

行解构造. 具体地,通过一个超参数select_percent $\in (0,1]$控制选取起点数量m,实现精度与速度的权衡. 其中$m=\lceil n\times \text{select_percent}\rceil$,从$n$个顶点中随机选取$m$个起点分别进行解码. 训练时,select_percent设为1,从而$m=n$,即所有的顶点各自作为起点构造解,以获取更多的解序列,加快模型学习.

2.2.2 自回归的构造解

选定起点π_1后,在$t(t\geqslant 1)$时刻,对起点嵌入$h^L_{\pi_1}$、当前解序列π的尾顶点嵌入$h^L_{\pi_t}$和图嵌入h_g进行拼接,并通过参数$W^{h_c}\in R^{d\times 3d}$做线性变换得到上下文嵌入$h_c$,有

$$h_c=W^{h_c}\text{concat}(h^L_{\pi_1},h^L_{\pi_t},h_g). \tag{8}$$

将h_c输入至解码器构造解序列,解码器由MHA层和顶点预测层构成. MHA层首先根据h_c构造查询向量c,根据顶点嵌入$h^L_i(i\in\{1,2,\dots,n\})$构造键向量$\text{key}_i$和值向量$v_i$,即

$$c=W^c h_c,\ \text{key}_i=W^{\text{key}}h^L_i,\ v_i=W^v h^L_i. \tag{9}$$

其中$W^c\in R^{d\times d},W^{\text{key}}\in R^{d\times d},W^v\in R^{d\times d}$为待学习的参数. 沿维度$d$方向将$c$、$\text{key}_i$和$v_i$拆分,分别得到num个子空间$c_e\in R^{\frac{d}{\text{num}}},\text{key}^e_i\in R^{\frac{d}{\text{num}}},v^e_i\in R^{\frac{d}{\text{num}}},e\in\{1,2,\dots,\text{num}\}$. 各子空间内根据式(10)计算注意力系数,对不符合条件的顶点进行Mask,即设置为$-\infty$,有

$$\text{ua}^e_{c,i}=\begin{cases}\dfrac{c_e^{\text{T}}\text{key}^e_i}{\sqrt{d_{\text{key}^e}}},\text{顶点}i\text{满足条件};\\ -\infty,\ \text{otherwise}.\end{cases} \tag{10}$$

其中:key^e_i为顶点i在第e个子空间的键向量,$d_{\text{key}^e}=\dfrac{d}{\text{num}}$为子空间的维度,$\text{ua}^e_{c,i}$为顶点$i$在第$e$个子空间下的注意力系数. 然后对$\text{ua}^e_{c,i}$进行softmax归一化,并与对应的值向量$v^e_i$加权求和,得到

$$\text{head}'_e=\text{softmax}(\text{ua}^e_{c,i})v^e_i. \tag{11}$$

不符合条件的顶点在经过softmax后数值为0,阻止其值向量信息输入到head'_e中. 多个子空间的结果合并后通过参数$W_f\in R^{d\times d}$做线性变换得到q_t,表示t时刻具有全局信息的查询向量,即

$$q_t=\text{concat}(\text{head}'_1,\dots,\text{head}'_{\text{num}})W_f. \tag{12}$$

顶点预测层将q_t分别与式(9)中key_i进行点积运算. 采用tanh函数和截断值clip调整数值区间为$[-\text{clip},\text{clip}]$,同样对不符合条件的顶点进行Mask,有

$$\text{unorm}_{q_t,i}=\begin{cases}\text{clip}\cdot\tanh\left(\dfrac{q_t^{\text{T}}\text{key}_i}{\sqrt{d}}\right),\text{顶点}i\text{满足条件};\\ -\infty,\ \text{otherwise}.\end{cases} \tag{13}$$

其中$\text{unorm}_{q_t,i}$为未进行归一化的概率数值. 对$\text{unorm}_{q_t,i}$做softmax运算后得到输出概率p_i,即

$$p_i=p_\theta(\pi_{t+1}=i|\pi_{1:t},s)=\text{softmax}(\text{unorm}_{q_t,i}), \tag{14}$$

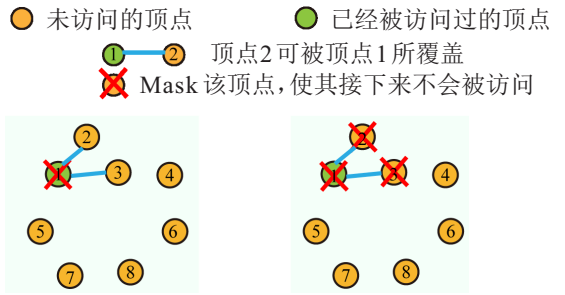
其中p_i为$t+1$时刻选择顶点i加入到解序列π的概率. 在训练阶段根据概率随机采样(sampling)选择顶点,而在测试阶段选择概率最大的顶点加入到解序列中,分别如下:

$$\pi_{t+1}=\text{sampling}(p_1,p_2,\dots,p_n), \tag{15}$$

$$\pi_{t+1}=\text{argmax}(p_1,p_2,\dots,p_n). \tag{16}$$

2.2.3 Mask策略

在自回归的构造解过程中,每个时刻进行节点预测时需要对不满足条件的节点进行Mask. AM-dynamic在求解CSP时选择只Mask访问过的节点,通过设计动态嵌入,希望模型能够学习到被访问过的顶点附近的顶点尽量不被选择这一特征. 但实验结果表明,当顶点数量大于100时,模型难以捕获CSP的问题特征,其最优间隙显著增加. 本文提出一种新的Mask策略,即对已经访问过的节点和其所能覆盖的节点均进行Mask. 该策略尽量多地减少待预测顶点的数目,特别在面对大规模实例时,每进行顶点选择时能显著缩小预测空间. 图6为AM-dynamic的Mask策略与本文所提出Mask策略的对比.



(a) AM-dynamic的Mask策略 (b) 本文Mask策略

图 6 两种策略对比

2.2.4 训练方法

根据模型解码时输出的概率分布$p_\theta(\pi|s)$获取完整的解序列π,将负路径长度作为奖励通过强化学习方法进行训练. 训练目标为最小化路径长度,定义损失Loss为所求解的CSP实例长度的期望,有

$$\text{Loss}(\theta|s)=E_{p_\theta(\pi|s)}\text{len}(\pi), \tag{17}$$

其中参数θ通过带基线(baseline)的REINFORCE算法利用梯度下降算法进行更新,即

$$\nabla_\theta\text{Loss}=E_{p_\theta(\pi|s)}[(\text{len}(\pi)-b(s))\nabla\log p_\theta(\pi|s)]. \tag{18}$$

基线$b(s)$的作用是用于减少由于采样的不确定性所产生的高方差,当策略的求解长度$\mathrm{len}(\pi)$高于基线时将进行抑制,低于基线则增强. 该基线可以是指数加权平均、评论家网络或greedy rollout[10]. 文献[15]提出了一个基于多轨迹序列的基线,即

$$b(s)=\frac{1}{\mathrm{multi}}\sum_{z=1}^{\mathrm{multi}}\mathrm{len}(\pi^z). \tag{19}$$

其中: multi为轨迹序列的个数,π^z为采样的第z个序列. 本文采用该基线进行训练,令$\mathrm{multi}=m$. 对于CSP问题实例s,从m个起点获取到m条路径长度的平均值作为基线,并使用Adam优化器进行参数更新. 特别地,通过多起点策略采样到多条轨迹的概率分布有助于进一步降低$\nabla\log p_\theta(\pi|s)$的方差,加速模型收敛. 此外,一些代码层面的优化,如利用GPU并行计算路径长度、覆盖顶点等,也加快了模型前向传播的运算速度.

3　实验设计

为验证所提出方法的有效性,设计如下实验: 1) 在CSP 20(顶点数量为20,下同)、CSP 50和CSP 100训练后与其他方法在测试集上的对比实验; 2) 在CSP 100训练的模型在CSP 200和CSP 300测试集上的泛化性对比实验. 3) 与训练时可覆盖集合数量不同的泛化性对比实验. 训练时设置每个顶点的可覆盖集合$\mathrm{Set}_i\ (i\in\{1,2,\dots,n\})$为距离其最近的NC个顶点[8],令$\mathrm{NC}=7$. 测试时为了验证模型对不同覆盖集合的泛化能力,每个顶点可覆盖集合的数目可以是任意的. 4) Mask策略和多起点方法的有效性分析实验.

3.1　对比的方法

所提出方法与传统启发式算法LS 1、LS 2和基于DNN的方法AM、AM-dynamic进行比较. 为确保公平性,实验在同一台机器上运行. GPU型号为2080Ti, CPU型号为Intel (R) Core (TM) i9-9900X CPU @ 3.50 GHz. 每训练一轮后在包含10 000个CSP实例的验证集上评估当前模型,并绘制学习曲线.

3.2　评价过程

将所有方法在包含1 000个CSP实例的测试集上进行测试,基于DNN的模型均采用式(16)构造解,评价指标为测试集的平均路径长度、最优间隙以及求解单个实例的运行时间,最优间隙由求解的路径长度$\mathrm{len}(\pi)$和已知的最优解$\mathrm{len}_{\mathrm{best}}$通过式(20)计算得到. 与AM-dynamic设置相同,本文同样在模型构造完整解的基础上使用简单局部搜索进一步改善解的质量,运行时间是上述方法所花费时间的总和,即

$$\text{最优间隙}=\frac{\mathrm{len}(\pi)-\mathrm{len}_{\mathrm{best}}}{\mathrm{len}_{\mathrm{best}}}. \tag{20}$$

4　实验结果

4.1　训练过程的比较

图7展示了AM-dynamic、AM和AM-NM在CSP 100实例上的学习曲线,其中AM-dy-32w为AM-dynamic每轮使用32万个训练样本. 在批次大小为64,每轮8万的训练样本数量条件下,AM-dynamic、AM的学习曲线震荡较为剧烈,而AM-NM的学习曲线十分平稳,同时收敛的最终结果也优于二者. 这一方面是由于针对CSP提出的Mask策略能够使网络快速地收敛到局部最优,另一方面是多起点策略提供了更多的采样路径,有助于降低梯度方差,加速模型训练. 此外,在给予更多的训练样本后,AM-dynamic的学习曲线趋于平稳,但最终的收敛结果仍然与AM-NM存在一定差距,这表明AM-NM有着更高的样本效率. 同时,AM-dynamic的训练时间为12 h而AM-NM的训练时间仅为4 h.

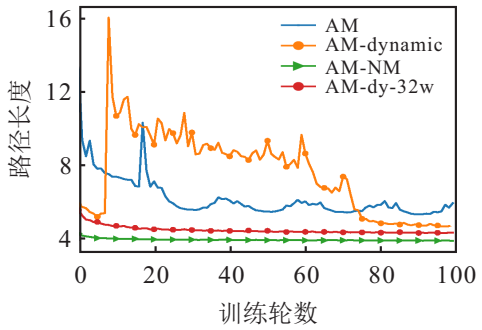


图 7　AM、AM-dynamic、AM-NM在CSP 100的学习曲线

4.2　在测试集的性能比较

表1给出了AM-NM与对比方法在CSP 20、CSP 50、CSP 100测试集上的实验结果. 与启发式算法相比,AM、AM-dynamic、AM-NM在求解速度上显著提升,但求解质量存在一定差距. 对比其他DNN方法,AM-NM能够取得更低的最优间隙. 表1第3部分展示了对AM-dynamic和AM-NM的解通过简单局部搜索进行改进后的结果,同时给出局部搜索在测试集中改善的实例个数,局部搜索改善的实例个数越少表明模型所构造的解的质量越高. 由于Mask策略显著地减少了可行动作空间,AM-NM的求解更接近局部最优解,对局部搜索的依赖程度更小,明显优于AM-dynamic. CSP 100的结果表明,随着顶点数量的增加,构造解的过程中面临着更多的可行动作空间,此时AM-NM构造局部最优解的能力有所减弱,但仍优于AM-dynamic. 后续的工作将考虑进一步提升AM-NM在大规模实例上的求解能力.

表 1 对比方法在CSP 20、CSP 50、CSP 100测试集上的实验结果

方法	CSP 20			CSP 50			CSP 100		
	路径长度	最优间隙/%	求解时间/s	路径长度	最优间隙/%	求解时间/s	路径长度	最优间隙/%	求解时间/s
LS 1	1.98	12.32	2.62	2.60	0.00	9.50	3.55	0.00	59.77
LS 2	1.76	0.00	4.41	2.68	2.95	16.41	3.70	4.25	80.15
AM	2.93	35.68	0.000	4.17	60.08	0.000	5.18	45.95	0.000
AM-dynamic	1.90	7.84	0.000	3.08	18.37	0.000	4.35	22.62	0.000
AM-NM	**1.85**	**4.77**	0.000	**2.73**	**4.74**	0.000	**3.79**	**6.69**	0.003
AM-dynamic (LS)	**1.83** (615)	**4.05**	0.32	2.80 (1 000)	7.71	0.48	3.63 (1 000)	2.38	2.17
AM-NM (LS)	1.85 (**4**)	4.76	0.32	**2.72** (**78**)	**4.53**	**0.38**	**3.57** (**997**)	**0.54**	**2.03**

5 结 论

本文针对基于深度强化学习的CSP求解方法中起点选择、Mask策略存在的不足做出改进研究,所提出的Mask策略在模型构造解过程中能够减少待预测顶点数目,多起点策略可以改善模型的训练过程并提高求解质量. 实验结果表明,所提出算法显著优于现有基于DNN的求解方法. 尽管所提出算法显著缩小了与启发式算法的最优间隙,但是仍然存在一定差距. 未来将尝试从解码策略、模型结构等方面继续提高求解质量.

参考文献(References)

[1] FLood M M. The traveling-salesman problem[J]. Operations Research, 1956, 4(1): 61-75.

[2] Current J R, Schilling D A. The covering salesman problem[J]. Transportation Science, 1989, 23(3): 208-213.

[3] Golden B, Naji-Azimi Z, Raghavan S, et al. The generalized covering salesman problem[J]. Informs Journal on Computing, 2012, 24(4): 534-553.

[4] Salari M, Naji-Azimi Z. An integer programming-based local search for the covering salesman problem[J]. Computers & Operations Research, 2012, 39(11): 2594-2602.

[5] Pandiri V, Singh A, Rossi A. Two hybrid metaheuristic approaches for the covering salesman problem[J]. Neural Computing and Applications, 2020, 32(19): 15643-15663.

[6] Salari M, Reihaneh M, Sabbagh M S. Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem[J]. Computers & Industrial Engineering, 2015, 83: 244-251.

[7] Tripathy S P, Tulshyan A, Kar S, et al. A metameric genetic algorithm with new operator for covering salesman problem with full coverage[J]. International Journal of Control Theory and Applications, 2017, 10(7): 245-252.

[8] Li K W, Zhang T, Wang R, et al. Deep reinforcement learning for combinatorial optimization: Covering salesman problems[J]. IEEE Transactions on Cybernetics, 2022, 52(12): 13142-13155.

[9] Vinyals O, Fortunato M, Jaitly N. Pointer networks[J/OL]. 2015, arXiv: 1506.03134.

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

[11] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在TSP优化中的应用[J]. 控制与决策, 2018, 33(2): 219-225. (He Q, Wu Y L, Xu T W. Application of improved genetic simulated annealing algorithm in TSP optimization[J]. Control and Decision, 2018, 33(2): 219-225.)

[12] Joshi C K, Cappart Q, Rousseau L M, et al. Learning the travelling salesperson problem requires rethinking generalization[J/OL]. 2020, arXiv: 2006.07054.

[13] 李凯文, 张涛, 王锐, 等. 基于深度强化学习的组合优化研究进展[J]. 自动化学报, 2021, 47(11): 2521-2537. (Li K W, Zhang T, Wang R, et al. Research reviews of combinatorial optimization methods based on deep reinforcement learning[J]. Acta Automatica Sinica, 2021, 47(11): 2521-2537.)

[14] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, 2017: 6000-6010.

[15] Kwon Y D, Choo J, Kim B, et al. POMO: Policy optimization with multiple optima for reinforcement learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 21188-21198.

作者简介

方伟(1980−), 男, 教授, 博士生导师, 从事智能优化算法、数据挖掘等研究, E-mail: fangwei@jiangnan.edu.cn;

接中冰(1999−), 男, 硕士生, 从事组合优化算法的研究, E-mail: 6201910024@stu.jiangnan.edu.cn;

陆恒杨(1991−), 男, 副教授, 博士生, 从事机器学习、自然语言处理等研究, E-mail: luhengyang@jiangnan.edu.cn;

张涛(1988−), 男, 高级工程师, 博士生，从事船舶结构性智能评价的研究, E-mail: zhangtao@cssrc.com.cn.