

基于图卷积和注意力神经网络的旅行商问题新解法

韦念念, 韩曙光

引用本文

韦念念, 韩曙光. [基于图卷积和注意力神经网络的旅行商问题新解法](#)[J]. 计算机科学, 2024, 51(6A): 230700222-8.

WEI Niannian, HAN Shuguang. [New Solution for Traveling Salesman Problem Based on Graph Convolution and Attention Neural Network](#) [J]. Computer Science, 2024, 51(6A): 230700222-8.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[结合图卷积神经网络和集成方法的推荐系统恶意攻击检测](#)

Malicious Attack Detection in Recommendation Systems Combining Graph Convolutional Neural Networks and Ensemble Methods

计算机科学, 2024, 51(6A): 230700003-9. <https://doi.org/10.11896/jsjcx.230700003>

[一种单阶段无监督可见光-红外跨模态行人重识别方法](#)

Single Stage Unsupervised Visible-infrared Person Re-identification

计算机科学, 2024, 51(6A): 230600138-7. <https://doi.org/10.11896/jsjcx.230600138>

[基于无监督显著性掩码引导的红外与可见光图像融合网络](#)

UMGN: An Infrared and Visible Image Fusion Network Based on Unsupervised Significance Mask Guidance

计算机科学, 2024, 51(6A): 230600170-5. <https://doi.org/10.11896/jsjcx.230600170>

[基于不变图卷积神经网络的文本分类](#)

Text Classification Based on Invariant Graph Convolutional Neural Networks

计算机科学, 2024, 51(6A): 230900018-5. <https://doi.org/10.11896/jsjcx.230900018>

[基于图卷积神经网络的点云语义分割综述](#)

Review of Point Cloud Semantic Segmentation Based on Graph Convolutional Neural Networks

计算机科学, 2024, 51(6A): 230400196-7. <https://doi.org/10.11896/jsjcx.230400196>

基于图卷积和注意力神经网络的旅行商问题新解法

韦念念 韩曙光

浙江理工大学理学院 杭州 310018

(wenn1998@163.com)

摘要 旅行商问题是一个经典的组合优化问题。为快速求解旅行商问题,设计了由图嵌入网络、图卷积神经网络、注意力神经网络和多层感知机组合而成的深度学习模型的学习分支规则,通过改进传统的分支定界算法提高算法性能。对15个城市的旅行商问题实例进行监督训练,并在SCIP求解器上分别测试10,15,20,25和30个城市的旅行商问题实例。发现:基于学习分支规则的分支定界算法的求解时间比基于传统分支规则的分支定界算法的求解时间分别快 -0.0022s 、 0.0178s 、 1.7643s 、 2.3074s 和 2.0538s 。因此,基于图神经网络的分支变量选择对传统分支规则的改进是有效的,可以较好地泛化到训练规模更大的旅行商问题实例中。

关键词 旅行商问题;图卷积神经网络;注意力网络;分支定界算法;监督学习

中图分类号 TP183

New Solution for Traveling Salesman Problem Based on Graph Convolution and Attention Neural Network

WEI Niannian and HAN Shuguang

School of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China

Abstract Traveling salesman problem is a classic combinatorial optimization problem. To solve the problem quickly, a learning branch rule is designed, which is based on a deep learning model composed of graph embedding network, graph convolutional neural network, attention neural network and multi-layer perceptron, and the traditional branch and bound algorithm is modified to improve the algorithm performance. Traveling salesman problem instances of 15 cities are supervised and trained, and the traveling salesman problem instances of 10, 15, 20, 25 and 30 cities are tested on the SCIP solver respectively. We find that the solution time of the branch and bound algorithm based on learning branch rule is -0.0022s , 0.0178s , 1.7643s , 2.3074s , and 2.0538s faster than that of the algorithm based on traditional branch rules, respectively. Therefore, the selection of branch variables based on graph neural networks is effective in improving traditional branch rules and can be well normalized to traveling salesman problem instances with larger training scales.

Keywords Traveling salesman problem, Graph convolutional neural network, Attention neural network, Branch and bound algorithm, Supervised learning

1 引言

旅行商问题^[1]是经典的组合优化问题,描述为:一个商品推销员要去若干个城市推销商品,该推销员从一个城市出发,需要经过所有城市后,回到出发地,应如何选择行进路线,以使总的行程最短。从图论角度看,实质是在一个带权完全无向图中,找一个权值最小的哈密顿回路。随着顶点数的增加,会产生组合爆炸。由于旅行商问题在交通运输^[2]、集成电路以及物流配送^[3]等领域有着广泛的应用,因此国内外学者对该问题进行了大量研究。

求解旅行商问题常用的精确算法包括分支定界法^[4]、分支切割法^[5]和动态规划法^[6]等,但随着问题规模的增大,精确算法会变得无能为力,因而近似算法和启发式算法不断涌现。其中,近似算法有近似比为 $1/2(\lg n + 1)$ 、时间复杂度为

$O(n^2 \lg n)$ 的最近邻算法^[7],近似比为 $2/7 \lg n + 5/21$ 、时间复杂度为 $O(n^2)$ 的Clark&Wright算法^[8],以及近似比为 $3/2$ 、时间复杂度为 $O(n^3)$ 的Christofides算法^[9]等。启发式算法有蚁群算法^[10]、粒子群算法^[11]、人工鱼群算法^[12]、混合蛙跳算法^[13]和人工蜂群算法^[14]等。

精确算法的计算量一般随问题规模的增大呈指数增长,其应用范围有限;近似算法能够保证最坏情况下解决方案的质量,但不存在足够大的界限,可扩展性比较局限;启发式算法需要设计有效的搜索规则,在求解问题时经常遇到搜索效率和求解精度之间的权衡问题。

另外,依据该问题在不同领域的应用,旅行商问题又衍生出许多扩展问题,如不对称旅行商问题^[15]、配送收集旅行商问题^[16]、多人旅行商问题^[17]、多目标旅行商问题^[18]、带有时间约束的旅行商问题^[19]和着色旅行商问题^[20]等。

基金项目:国家自然科学基金(12071436)

This work was supported by the National Natural Science Foundation of China(12071436).

通信作者:韩曙光(zist001@163.com)

2013年,深度神经网络兴起,机器学习便被应用于旅行商问题并取得满意的结果。目前,基于机器学习的旅行商问题解法主要分为两类。

第一类是使用机器学习模型直接从旅行商问题定义中获得解决方案。Vinyals等^[21]提出了一种基于长短时记忆网络的序列到序列的模型——指针网络,采取监督训练的方式并自回归地输出城市节点排列。Bello等^[22]使用演说者-评论家算法训练指针网络。该算法视每个实例为训练样本,使用样本的路径长度对策略梯度进行无偏蒙特卡洛估计。Kool等^[23]将文献[22]的训练方法应用Transformer,该架构是可用于完整图的图注意力网络。Nowak等^[24]有监督地训练一个图神经网络,输出路径的邻接矩阵并使用光束搜索将其转换为可行解。Joshi等^[25]训练一个图卷积神经网络生成旅行商问题热图,使用文献[24]的搜索方式构造有效路径。Kool等^[26]将文献[25]的研究扩展到车辆路径问题上,使用动态规划算法构建路径。Fu等^[27]训练一个图神经网络生成小规模旅行商问题热图,通过子图采样构建大规模旅行商问题热图,采取强化学习训练方式并使用蒙特卡洛搜索构造有效路径。

第二类是使用机器学习模型获取某些数学信息,增强可解决旅行商问题的传统优化算法。Deudon等^[28]采用策略梯度训练一个基于Transformer的网络,使用2-opt搜索构建可行路径。Dai等^[29]对旅行商问题实例进行图神经网络编码,使用深度Q-learning网络训练一个structure2vec的图嵌入网络,采取最便宜的插入启发式构造有效路径。Wu等^[30]和Da-costa等^[31]均提出策略梯度算法,学习将2-opt应用于现有可行解决方案的策略。但文献[30]使用的网络是Transformer,文献[31]使用的网络是图卷积神经网络与循环神经网络的组合,后者获得了更好的实验结果。

基于机器学习的旅行商问题解法是一种新的数据驱动解决方案,通过模仿最佳求解器或通过强化学习来学习这些启发式和规则,将更多深度学习的思想吸收融入问题中,补充传统优化算法,使问题的求解过程变得高效。

本文试图使用机器学习模型获取变量的分支信息,增强分支定界算法并加快旅行商问题的求解。

分支定界算法是一种用于产生无法在多项式时间内解决的组合优化问题的精确解的方法^[32],被广泛应用于旅行商问题^[33]、机器调度问题^[34]、图划分问题^[35]和二次指派问题^[36]等。Land等^[37]首次提出使用分支定界算法解决离散规划问题。该算法通过迭代将原始问题划分为一系列子问题,生成树的结构并丢弃无法找到全局最优的子问题,从而隐式列举了所有可能的解决方案。Clausen等^[38]指出变量选择(即分支哪个分数变量)、节点选择(即接下来要探索哪个当前节点)、修剪规则(即阻止探索次优空间的规则)和切割规则(即为查找切割平面添加约束的规则)等决策影响着分支定界算法性能。有效的决策可以缩小解空间的大小并加快搜索最优解的速度。

而强分支是一种重要的变量选择规则,它通过求解线性松弛问题来预测潜在子节点的上界与下界,从而保持分支定界树的结构较小^[39]。Gamrath等^[40]表明通过将域传播纳入强分支来改进双界预测可以减少每次强分支调用的平均线性松弛迭代次数、分支定界节点的数量以及整体求解时间。

Kilinc等^[41]通过基于分支的线性化、析取不等式和混合型不等式等形式的强分支生成信息来加强混合整数非线性规划问题的松弛。Santanu等^[42]使用顶点覆盖实例证明强分支规则是有效的,且与所有其他已知的变量选择规则相比,该规则可以产生更小的分支定界树。同时,基于强分支的分支定界树的大小在最佳分支定界树大小的两倍以内,但计算成本巨大。

为克服强分支规则昂贵的计算成本,需要在分支规则的质量与分支规则的成本之间寻找平衡。为此,我们使用机器学习模型学习一个能够快速逼近强分支规则的分支函数,指导分支变量的选择和旅行商问题的求解。

本文第2章描述了旅行商问题及其数学模型;第3章重点介绍图神经网络模型的构造、搭建、训练与应用;第4章展示了算法对比的实验结果;最后总结全文并展望未来。

2 问题描述与数学模型

旅行商问题可描述为:一个商品推销员要去若干个城市推销商品,该推销员从一个城市出发,需要经过所有城市后,回到出发地,应如何选择行进路线,以使总的行程最短。

旅行商问题的数学模型^[43]如下:

$$\begin{aligned} \min & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ & \sum_{i=1, i \neq j}^n x_{ij} = 1, \forall j \in \{1, \dots, n\} \\ & \sum_{j=1, j \neq i}^n x_{ij} = 1, \forall i \in \{1, \dots, n\} \\ & u_i - u_j + n x_{ij} \leq n - 1, \forall 2 \leq i \neq j \leq n \\ & 0 \leq u_i \leq n - 1, \forall i \in \{2, \dots, n\} \\ & x_{ij} \in \{0, 1\}, \forall i, j \in \{1, \dots, n\} \end{aligned}$$

其中, c_{ij} 表示城市*i*与城市*j*之间的欧氏距离; x_{ij} 表示商品推销员是否在访问城市*i*之后立即访问城市*j*; u_j 表示城市*j*的访问次序。

3 基于图模型的算法设计

分支定界算法是一种被广泛应用于求解旅行商问题的精确算法。该算法的核心是分支变量选择^[44]:根据不同的分支规则计算所有候选变量的得分,选择分数最高的变量进行分支。最有效的分支规则之一是强分支规则,但该规则的计算成本巨大。因此,我们使用机器学习模型来学习一个能够快速逼近强分支规则的分支函数,获取变量的分支信息,指导分支变量的选取,从而加快旅行商问题的求解。本文方法将分支定界树当前关注节点的线性松弛问题构建为二部图,分支函数编码为图神经网络,最小化交叉熵进行监督训练,快速逼近表现良好的强分支规则。

3.1 图模型的构造

将分支定界树当前关注节点的线性松弛问题 LP_i 构建为二部图 $G=(\mathcal{V}, \mathcal{E})$,如图1所示。图的顶点由变量顶点集合 \mathcal{V}_V 和约束顶点集合 \mathcal{V}_C 组成,集合 \mathcal{V}_V 中的每一个变量顶点 v 对应 LP_i 的一个决策变量,集合 \mathcal{V}_C 的每一个约束顶点 c 对应 LP_i 的一个约束。如果变量顶点 v 在约束顶点 c 中所对应的系数不为0,那么在变量顶点 v 与约束顶点 c 之间便存在一条变量-约束边,且约束系数为该边的特征。

表1和表2分别描述了变量顶点和约束顶点的特征的符号及其说明。

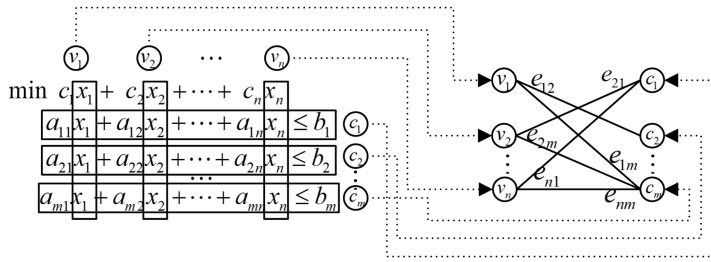


图 1 线性松弛问题的二部图表示

Fig. 1 Bipartite graph representation of linear relaxation problem

表 1 变量顶点特征的符号及其说明

Table 1 Symbols and descriptions of variable vertex features

符号	说明
objective	表示变量在目标函数中的系数
is_type_binary	表示变量是否是二进制变量,即取值只能是 0 或 1
is_type_integer	表示变量是否是整数变量,即取值只能是整数
is_type_implicit_integer	表示变量是否是隐式整数变量,即虽然类型被定义为连续型变量,但是由于取值范围限制在整数集合中,可以被视为整数变量
is_type_continuous	表示变量是否是连续型变量,即可以取任何实数值
has_lower_bound	表示变量是否有下界限制
has_upper_bound	表示变量是否有上界限制
normed_reduced_cost	表示变量的归一化成本,用于解释变量对目标函数的贡献程度
solution_value	表示变量在最优解中的取值
solution_fraction	表示变量在最优解中的分数部分,即小数点后的部分
is_solution_at_lower_bound	表示变量在最优解中是否达到了下界限制
is_solution_at_upper_bound	表示变量在最优解中是否达到了上界限制
scaled_ageincumbent_value	表示变量在最近若干次迭代中的平均取值
average_incumbent_value	表示变量在所有已知最优解中的平均取值
is_basis_lower	表示变量是否是基变量的下界,即在单纯形法中是否是基变量的下界

表 2 约束顶点特征的符号及其说明

Table 2 Symbols and descriptions of constraint vertex features

符号	说明
bias	表示约束条件和目标函数的偏移量
objective_cosine_similarity	表示目标函数向量和约束条件向量之间的余弦相似度,用来判断目标函数是否收敛
is_tight	表示约束条件是否为紧约束
dual_solution_value	表示对偶问题的目标函数的取值,通过计算对偶问题的解,可以得到原问题的最优解
scaled_age	表示当前解的质量,值越小,表示当前解越优

3.2 图模型的搭建

为学习强分支规则,我们将分支函数编码为图神经网络模型,如算法 1 所示。该图模型可分为 5 个阶段:1)预正则化层:消除顶点与边的特征的平均值和标准差,避免出现过拟合;2)嵌入层:使顶点与边的特征具有相同的维数;3)图卷积层:通过邻居顶点与边进行顶点的信息聚合和传递;4)注意力机制层:通过邻居变量顶点进行变量顶点的信息聚合和传递;5)多层感知机层:作用于每个变量顶点并输出它们的得分。

算法 1 图神经网络模型算法

输入:图 $G=(\mathcal{V}, \mathcal{E})$; 输入特征 $x_j, \forall j \in \mathcal{V}$; 嵌入层权重矩阵 \mathbf{W}^{emb} ; 图卷积层权重矩阵 $\mathbf{W}^{\text{con}_1}$ 和 $\mathbf{W}^{\text{con}_2}$; 邻域函数 \mathcal{N} ; 注意力系数矩阵 \mathbf{A} ; 输出层权重矩阵 \mathbf{W}^{out} ; 非线性函数 σ

输出:变量得分 $y_v, \forall v \in \mathcal{V}$

1. $h_v \leftarrow \sigma(\mathbf{W}_V^{\text{emb}} \cdot \text{BN}(x_v)), \forall v \in \mathcal{V}$

2. $h_c \leftarrow \sigma(\mathbf{W}_C^{\text{emb}} \cdot \text{BN}(x_c)), \forall c \in \mathcal{C}$

3. $h_e \leftarrow \sigma(\mathbf{W}_E^{\text{emb}} \cdot \text{BN}(x_e))$

4. for c in \mathcal{C} do:

$h'_c \leftarrow \sigma(\mathbf{W}_{V_c}^{\text{con}_1}, \text{concat}(h_c, h_e))$

$h_c \leftarrow \sigma(\mathbf{W}_{V_c}^{\text{con}_1} \cdot \text{concat}(h'_c, \sum_{v \in \mathcal{V} \cap \mathcal{N}(c)} w_{vc} h_v))$

5. for v in \mathcal{V} do:

$h'_v \leftarrow \sigma(\mathbf{W}_{e_v}^{\text{con}_2} \cdot \text{concat}(h_c, h_v))$

$h_v \leftarrow \sigma(\mathbf{W}_{e_v}^{\text{con}_2} \cdot \text{concat}(h'_v, \sum_{c \in \mathcal{C} \cap \mathcal{N}(v)} w_{vc} h_c))$

6. for v in \mathcal{V} do:

$$h_v \leftarrow \sigma(h_v, \sum_{j \in \{\mathcal{V}, \mathcal{V}\}} \Lambda_{vj} h_j)$$

$$7. y_v \leftarrow \sigma(\mathbf{W}^{\text{out}} \cdot h_v), \forall v \in \mathcal{V}$$

在二部图中,顶点的特征信息根据以下两个交错的半卷积层进行更新,如图 2 所示。

1)约束顶点的特征 h_c 由它的邻居变量顶点的特征 h_v 和与之相关联的边的特征 h_e 进行更新。

2)变量顶点的特征 h_v 由它的邻居约束顶点的特征 h_c 和与之相关联的边的特征 h_e 进行更新。

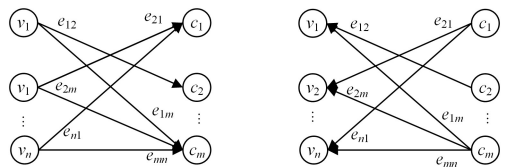


图 2 图卷积层的信息流转换示意图

Fig. 2 Schematic diagram of information flow transfer in graph convolutional layers

通过上述图卷积层可知,一个变量顶点可以聚合来自于与该顶点相关联的约束顶点和边的更多信息。为增强变量顶点之间的信息关联,在图卷积层后便引入了一个注意力机制层。最后,将多层感知机应用于变量顶点,生成候选分支变量的得分,选择分数最高的变量进行分支。整个神经网络模型的结构搭建如图 3 所示。

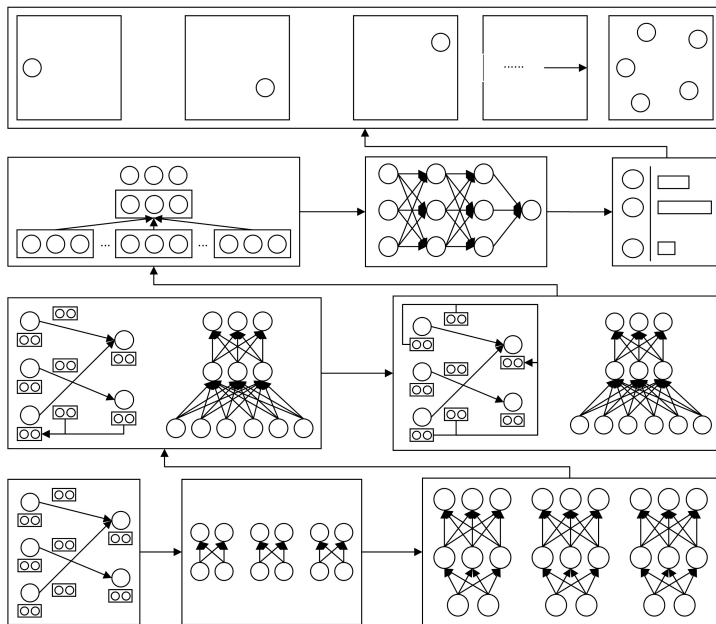


图3 图神经网络模型的结构示意图

Fig. 3 Schematic diagram of the structure of neural network model

3.3 图模型的训练

根据损失函数最小,使用 Adam 优化算法对网络参数进行更新,如图 4 所示。

其中损失函数是交叉熵函数:

$$L(\theta) = -\frac{1}{N} \sum_{(LP, \mathcal{V}_V^*) \in D} \log \pi_{\theta}(\mathcal{V}_V^* | LP)$$

这里, $D = \{(LP_i, \mathcal{V}_V^*)\}_{i=1}^N$ 是随机生成的旅行商问题实例在规划求解器中使用强分支规则进行分支定界时所记录的线性松弛问题一被选择的分支变量标签数据对集合。

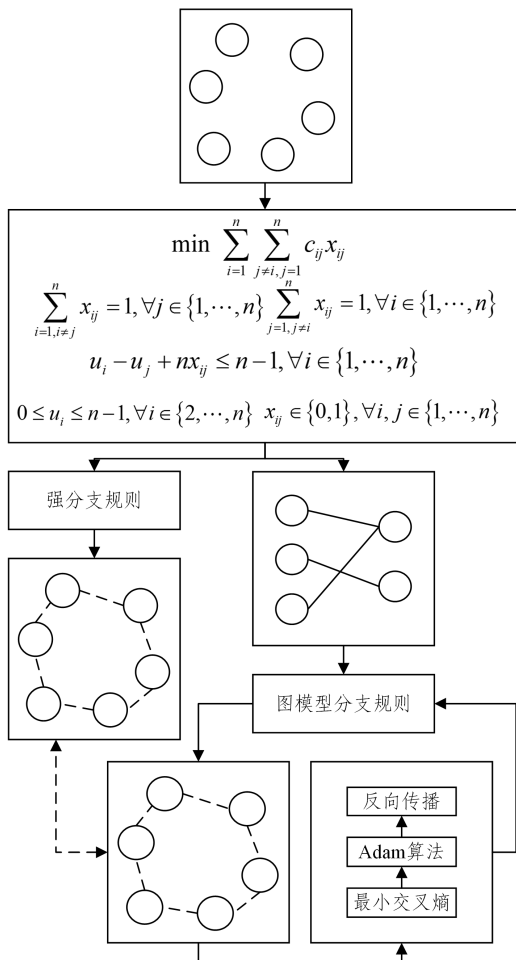


图4 图神经网络模型的训练示意图

Fig. 4 Schematic diagram of the training of neural network model

3.4 基于图模型的分支定界算法

对于每个待解决的旅行商问题实例,我们使用基于图模型学习分支规则的分支定界算法进行求解,如算法 2 和图 5 所示。

算法 2 基于图模型的分支定界算法

输入:旅行商问题的数学模型

输出:最优解 $x^* \in X_{MIP}$, 最优解值 $c^* = c^T x^*$ (x^* 是旅行商问题的最优行驶路径, c^* 是该路径的总行驶距离)

1. 旅行商问题的数学模型写成混合整数规划的一般形式: $c^* = \min c^T x, Ax \leq b, x \in Z$;
2. 初始化旅行商问题的数学模型的线性松弛问题集合 $S := \{PLP\}$ 。令 $c^* = \infty$;
3. 若 $S = \emptyset$, 则退出算法并返回最优解 x^* 和最优解值 c^* ;
4. 选择一个松弛问题 $Q \in S$, 更新 $S = S \setminus Q$;
5. 求解线性规划 $\min\{c^T x | x \in Q\}$, 得到最优解 \bar{x}_Q 和最优解值 \bar{c}_Q ;
6. 若 $\bar{c}_Q \geq c^*$, 则转到 3;
7. 若 $\bar{x}_Q \in X_{MIP}$, 则令 $c^* := \bar{c}_Q, x^* := \bar{x}_Q$, 并转到 3; (X_{MIP} 表示具有一般形式的混合整数规划的可行解集合)
8. 分支:

8.1. 分支变量选择

输入:当前子问题 Q 和最优松弛解 $\bar{x}_Q \notin X_{MIP}$

输出:分数变量 $\bar{x}_{Qi} \notin Z$ 的下标 i

- 8.1.1. 令 $C = \{i \in I | \bar{x}_{Qi} \notin Z\}$ 为分支候选变量集合;
- 8.1.2. 对所有候选变量 $i \in C$, 根据基于图模型学习的分支规则计算得分 $s_i \in R$;
- 8.1.3. $s_i = \max_{j \in C} \{s_j\}$, 并返回下标 i ;
- 8.2. 把问题 Q 分解成分别添加约束 $x_i \leq \lfloor \bar{x}_{Qi} \rfloor$ 和约束 $x_i \geq \lceil \bar{x}_{Qi} \rceil$ 的两个子问题, 并把它们加到集合 S 中, 转到 4。

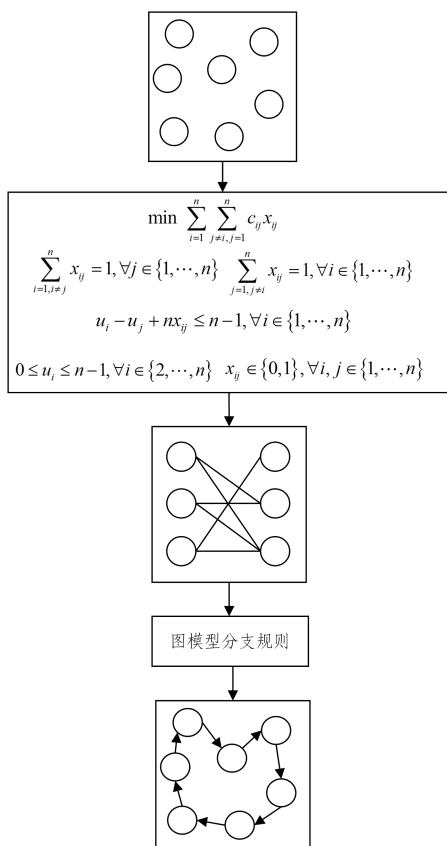


图 5 图神经网络模型的应用示意图

Fig. 5 Schematic diagram of the application of neural network model

4 数值实验

首先,对随机产生的具有 15 个城市的旅行商问题实例进行监督训练,其中训练样本数为 10 000,验证样本数为 2 000。然后,分别在标准 SCIP 求解器和基于图神经网络的分支规则的 SCIP 求解器上测试规模为 10, 15, 20, 25 和 30 个城市的旅行商问题实例(因为 35 个城市的旅行商问题测试时间超过 36 000 s, 计算成本过高, 所以暂不考虑 30 个以上城市的旅行商问题实例), 其中每种规模的测试样本数均为 50, 结果如表 3—表 7 所列。最后, 对比每个规模实例在不同求解器上的平均求解时间, 如图 6 所示。所有的实验均在机器 RTX 3080 GPU 和 Intel Xeon Platinum 8255C CPU @ 2.50 GHz 上实现。

表 3 基于学习分支规则的分支定界算法和基于传统分支规则的分支定界算法求解 50 个规模为 10 个城市的旅行商问题实例的时间
Table 3 Solution time of 50 traveling salesman problem instances in 10 cities using the branch and bound algorithm based on learning branch rules and traditional branch rules (s)

模型	求解时间				
GNN	0.24	0.23	0.29	0.20	0.27
SCIP	0.25	0.19	0.30	0.22	0.23
GNN	0.17	0.17	0.32	0.19	0.23
SCIP	0.22	0.23	0.23	0.20	0.21
GNN	0.37	0.32	0.30	0.69	0.26
SCIP	0.37	0.33	0.31	0.63	0.23
GNN	0.26	0.24	0.42	0.19	0.21
SCIP	0.24	0.20	0.42	0.17	0.31
GNN	0.31	0.19	0.28	0.23	0.30
SCIP	0.29	0.22	0.36	0.23	0.30
GNN	0.22	0.29	0.17	0.27	0.29
SCIP	0.24	0.33	0.20	0.24	0.32
GNN	0.24	0.27	0.27	0.16	0.30
SCIP	0.20	0.25	0.25	0.15	0.27
GNN	0.26	0.16	0.31	0.25	0.20
SCIP	0.25	0.17	0.31	0.25	0.19
GNN	0.25	0.31	0.26	0.19	0.24
SCIP	0.26	0.23	0.26	0.20	0.26
GNN	0.25	0.17	0.31	0.25	0.19
SCIP	0.23	0.17	0.25	0.22	1.26

注:0.24 表示基于学习分支规则的分支定界算法(GNN)求解第 1 个实例所用的时间;0.25 表示基于传统分支规则的分支定界算法(SCIP)求解第 1 个实例所用的时间。同理给出基于上述两种算法求解剩余 49 个实例所用的时间。

表 4 基于学习分支规则的分支定界算法和基于传统分支规则的分支定界算法求解 50 个规模为 15 个城市的旅行商问题实例的时间

Table 4 Solution time of 50 traveling salesman problem instances in 15 cities using the branch and bound algorithm based on learning branch rules and traditional branch rules (s)

模型	求解时间				
GNN	4.83	0.55	0.80	2.90	0.51
SCIP	4.85	0.60	0.86	2.12	0.76
GNN	1.04	8.50	0.61	2.03	1.19
SCIP	0.86	8.77	1.10	1.39	1.17
GNN	2.42	2.25	0.51	0.76	6.27
SCIP	3.02	2.34	0.64	0.78	6.07
GNN	1.03	1.16	2.45	6.58	2.87
SCIP	0.77	1.20	3.07	6.62	2.94
GNN	0.53	4.90	1.37	1.04	0.66
SCIP	0.55	4.93	1.16	1.06	0.66

(续表)

模型	求解时间				
GNN	0.71	0.83	1.06	1.08	1.42
SCIP	0.58	0.70	1.01	1.08	1.45
GNN	12.15	1.00	1.23	1.80	1.87
SCIP	13.13	1.44	1.30	1.72	1.86
GNN	1.22	0.81	0.81	0.90	0.75
SCIP	1.99	0.93	0.72	0.94	0.74
GNN	1.50	4.72	1.61	2.20	4.24
SCIP	1.29	4.55	1.71	2.12	3.85
GNN	0.80	0.63	3.16	3.96	0.82
SCIP	0.75	0.84	1.71	4.20	1.03

注:4.83表示基于学习分支规则的分支定界算法(GNN)求解第1个实例所用的时间;4.85表示基于传统分支规则的分支定界算法(SCIP)求解第1个实例所用的时间。同理给出基于上述两种算法求解剩余49个实例所用的时间。

表5 基于学习分支规则的分支定界算法和基于传统分支规则的分支定界算法求解50个规模为20个城市的旅行商问题实例的时间

Table 5 Solution time of 50 traveling salesman problem instances in 20 cities using the branch and bound algorithm based on learning branch rules and traditional branch rules

(s)

模型	求解时间				
GNN	11.79	15.71	4.72	19.42	6.50
SCIP	11.79	15.88	4.68	19.25	6.17
GNN	45.36	15.38	40.22	10.41	4.66
SCIP	35.90	15.72	39.89	10.49	4.76
GNN	11.28	29.80	13.27	6.71	90.73
SCIP	11.02	30.00	8.68	6.91	97.38
GNN	6.50	10.86	57.06	12.81	850.78
SCIP	6.42	10.76	42.06	13.13	957.43
GNN	4.80	18.26	48.89	9.02	6.82
SCIP	3.01	10.41	51.66	8.90	6.74
GNN	15.85	3.90	3.12	1.66	6.29
SCIP	15.26	6.75	2.84	1.71	5.43
GNN	4.58	5.81	35.58	23.67	28.01
SCIP	4.41	4.91	35.73	24.03	21.16
GNN	23.96	15.05	3.11	5.26	477.28
SCIP	23.45	15.09	3.02	4.19	470.54
GNN	14.74	25.28	10.38	12.66	1.41
SCIP	14.88	25.05	18.98	18.04	1.40
GNN	183.13	4.76	4.39	14.20	4.36
SCIP	186.28	4.88	3.62	23.20	4.48

注:11.79表示基于学习分支规则的分支定界算法(GNN)求解第1个实例所用的时间;11.79表示基于传统分支规则的分支定界算法(SCIP)求解第1个实例所用的时间。同理给出基于上述两种算法求解剩余49个实例所用的时间。

表6 基于学习分支规则的分支定界算法和基于传统分支规则的分支定界算法求解50个规模为25个城市的旅行商问题实例的时间

Table 6 Solution time of 50 traveling salesman problem instances in 25 cities using the branch and bound algorithm based on learning branch rules and traditional branch rules

(s)

模型	求解时间				
GNN	178.19	1330.64	18.04	45.47	62.83
SCIP	158.30	1670.38	17.86	45.41	63.01
GNN	119.19	44.70	60.87	6.33	36.01
SCIP	118.28	45.10	60.06	6.47	33.67
GNN	76.18	58.45	21.26	116.91	55.88
SCIP	76.10	58.16	21.23	118.03	55.64
GNN	407.43	1924.25	70.80	48.65	30.22
SCIP	298.84	1932.76	71.34	48.68	30.22
GNN	44.16	46.12	17.10	62.42	38.32
SCIP	43.88	53.20	15.64	115.32	38.60

(续表)

模型	求解时间				
GNN	39.64	74.09	255.12	146.44	74.89
SCIP	39.54	73.94	256.43	145.75	74.61
GNN	44.61	21.92	25.36	162.09	14.75
SCIP	44.97	22.17	25.33	162.57	14.57
GNN	24.85	34.85	506.83	94.34	31.36
SCIP	16.67	24.23	444.64	95.54	31.34
GNN	36.70	110.10	100.43	118.66	22.51
SCIP	35.89	110.04	76.23	119.00	22.72
GNN	32.93	11.85	16.70	83.89	13.25
SCIP	32.88	12.22	16.97	25.63	13.34

注:178.19表示基于学习分支规则的分支定界算法(GNN)求解第1个实例所用的时间;158.3表示基于传统分支规则的分支定界算法(SCIP)求解第1个实例所用的时间。同理给出基于上述两种算法求解剩余49个实例所用的时间。

表7 基于学习分支规则的分支定界算法和基于传统分支规则的分支定界算法求解50个规模为30个城市的旅行商问题实例的时间

Table 7 Solution time of 50 traveling salesman problem instances in 30 cities using the branch and bound algorithm based on learning branch rules and traditional branch rules

(s)

模型	求解时间				
GNN	375.06	39.37	457.16	1304.88	1996.07
SCIP	377.25	39.65	455.09	1320.16	2010.95
GNN	202.56	72.58	2036.10	34.84	1365.32
SCIP	202.65	72.92	2027.35	34.84	1365.32
GNN	139.02	130.19	336.40	614.02	226.54
SCIP	139.46	130.07	334.92	611.78	226.23
GNN	2199.76	63.40	88.36	3093.48	221.47
SCIP	2193.20	63.30	88.16	3087.88	222.47
GNN	843.11	40.58	39.06	448.27	1976.87
SCIP	842.09	40.65	38.69	445.19	1962.58
GNN	6624.79	2108.76	148.15	221.28	621.54
SCIP	6694.88	2120.30	148.52	220.40	617.95
GNN	186.46	177.05	547.10	278.28	695.98
SCIP	186.30	177.89	542.43	277.99	702.40
GNN	234.21	125.09	374.19	110.53	206.64
SCIP	234.35	125.68	372.56	109.87	205.59
GNN	580.09	60.66	105.63	494.97	6189.83
SCIP	580.43	60.81	104.83	495.03	6254.22
GNN	6542.95	177.13	146.64	217.78	612.96
SCIP	6520.66	176.76	146.18	218.02	608.95

注:375.06表示基于学习分支规则的分支定界算法(GNN)求解第1个实例所用的时间;377.25表示基于传统分支规则的分支定界算法(SCIP)求解第1个实例所用的时间。同理给出基于上述两种算法求解剩余49个实例所用的时间。

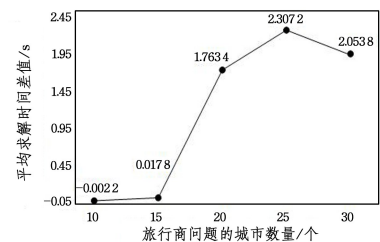


图6 基于学习分支规则的分支定界算法和基于传统分支规则的分支定界算法求解不同规模旅行商问题的平均所用时间差值
Fig. 6 Average solution time difference of traveling salesman problems of different scales using the branch and bound algorithm based on traditional branch rules and learning branch rules

结果表明,对于每种规模的实例而言,后者的平均求解时间要比前者分别快-0.0022s,0.0178s,1.7643s,2.3074s和2.0538s。说明基于图神经网络的分支变量选择策略对传统

分支规则的改进是有效的,而且可以很好地范化到比训练中使用的更大规模的旅行商问题实例中。同时我们分析:图模型需要经过训练数据的学习和参数调整过程,可能无法在相对较小规模问题上充分发挥其优势,导致其在具有 10 个城市的旅行商问题实例上测试效果不佳。而对于相对较大规模问题,如具有 10,15,20,25 和 30 个城市的旅行商问题实例,该学习模型的学习能力和推断能力能够使其在更短的时间内提供最优的解决方案。

结束语 本文对求解旅行商问题的传统算法——分支定界算法进行改进。我们的方法是将分支状态表示为二部图,使用一组特征来描述该状态,构建有监督的图神经网络来近似逼近强分支规则函数,并在随机生成的旅行商问题实例上应用基于学习的分支规则,最终能够在相对较短的时间内获得最优路径。但存在的问题是基于学习的分支规则的分支定界算法只能求解中小规模的旅行商问题,其应用范围具有一定的局限性。

未来,我们可以考虑使用图神经网络研究适用于旅行商问题的其他变体的分支策略,如车辆路径问题;同时,分支决策的质量与时间需权衡,可以根据问题背景来设计两者相平衡的分支策略,如混合分支等。

参 考 文 献

[1] PAPADIMITRIOU C H. The Euclidean travelling salesman problem is NP-complete [J]. *Theoretical Computer Science*, 1977,4(3):237-244.

[2] NOGAREDA A M, SER D J, OSABA E, et al. On the design of hybrid bio-inspired meta-heuristics for complex misattribute vehicle routing problems [J]. *Expert Systems*, 2020, 37(6): e12528.

[3] NOVELLANI S. Models and algorithms for the optimization of real-world routing and logistics problems [J]. *A Quarterly Journal of Operations Research*, 2016,14:331-332.

[4] DANTZIG G, FULKERSON R, JOHNSON S. Solution of a Large-Scale Traveling Salesman Problem [J]. *Operations Research*, 1954,2(4):365-462.

[5] FISCHETTI M, GONZÁLEZ J J S, TOTH P. A Branch-and-Cut Algorithm for the Symmetric Generalized Traveling Salesman Problem [J]. *Operations Research*, 1997,45(3):327-494.

[6] BELLMAN R. Dynamic Programming Treatment of the Traveling Salesman Problem [J]. *Journal of the ACM*, 1962,9(1):61-63.

[7] TASSIULAS L. Worst Case Length of Nearest Neighbor Tours for the Euclidean Traveling Salesman Problem [J]. *SIAM Journal on Discrete Mathematics*, 1997,10(2):171-179.

[8] CLARKE G, WRIGHT J W. Scheduling of vehicles from a central depot to a number of delivery points [J]. *Operations Research*, 1964,12(4):519-643.

[9] CHRISTOFIDES N. Worst-case analysis of a new heuristic for the travelling salesman problem [J]. *Operations Research Forum*, 2022,3(1):20.

[10] DORIGO M, GAMBARDILLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. *IEEE Transactions on Evolutionary Computation*, 1997,1(1):53-66.

[11] CHEN Y, ZHANG Y D, JING L, et al. An integer-coded chaotic

particle swarm optimization for traveling salesman problem [M]// *Progress in Robotics*. Berlin: Springer, 2009:372-379.

[12] ZHU M W, SHE X Y. Improved artificial fish swarm algorithm for solving traveling salesman problems [J]. *Computer Application Research*, 2010,27(10):3734-3736.

[13] LI Z F, WANG Y H. An improved shuffled frog leaping algorithm for TSP [M]// *Advances in Multimedia, Software Engineering and Computing Vol. 2*. Berlin: Springer, 2012:139-144.

[14] KARABOGA D, GORKEMLI B. A combinational artificial bee colony algorithm for traveling salesman problem [C]// *International Symposium on Innovations in Intelligent Systems and Applications*. 2011:50-53.

[15] CARPANETO G, DELL'AMICO M, TOTH P. Exact solution of large-scale, asymmetric traveling salesman problems [J]. *ACM Transactions on Mathematical Software*, 1995,21(4):394-409.

[16] GOETSCHALCKX M, JACOBS-BLECHA C. The vehicle routing problem with backhauls [J]. *European Journal of Operational Research*, 1989,42(1):39-51.

[17] XUE M H, WANG T Z, MAO S. Double evolutionary artificial bee colony algorithm for multiple traveling salesman problem [J]. *MATEC Web of Conferences*, 2016,44:02025.

[18] BOUZOUBÍA S, LAYEB A, CHIKHI S. Enhanced Chemical Reaction Optimization for Multi-objective Traveling Salesman Problem [M]// *Modelling and Implementation of Complex Systems*. Springer, 2016:91-106.

[19] MAUTOR T, NAUDIN E. Arcs-states models for the vehicle routing problem with time windows and related problems [J]. *Computers & Operations Research*, 2007,34(4):1061-1084.

[20] LI J, ZHOU M C, SUN Q R, et al. Colored traveling salesman problem [J]. *IEEE Transactions on Cybernetics*, 2015,45(11):2390-2401.

[21] VINYALS O, FORTUNATO M, JAITLY N. Pointer networks [C]// *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS' 15)*. 2015:2692-2700.

[22] BELLO I, PHAM H, QUOC V L, et al. Neural combinatorial optimization with reinforcement learning [J]. *arXiv*:1611.09940, 2016.

[23] KOOL W, HOOF H V, WELLING M. Attention, learn to solve routing problems [J]. *arXiv*:1803.08475, 2018.

[24] NOWAK A, VILLAR S, BANDEIRA A S, et al. Revised note on learning algorithms for quadratic assignment with graph neural networks [J]. *arXiv*:1706.07450, 2018.

[25] JOSHI C K, LAURENT T, BRESSON X. An efficient graph convolutional network technique for the travelling salesman problem [J]. *arXiv*:1906.01227, 2019.

[26] KOOL W, HOOF H V, GROMICHO J, et al. Deep policy dynamic programming for vehicle routing problems [M]// *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 2022:190-213.

[27] FU Z H, QIU K B, ZHA H Y. Generalize a small pre-trained model to arbitrarily large tsp instances [C]// *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021:7474-7482.

[28] DEUDON M, COURNUT P, LACOSTE A, et al. Learning heuristics for the TSP by policy gradient [C]// *Integration of Con-*

- straint Programming, Artificial Intelligence, and Operations Research, 2018:170-181.
- [29] DAI H, KHALIL E B, ZHANG Y Y, et al. Learning combinatorial optimization algorithms over graphs [C] // Proceedings of the 31st International Conference on Neural Information Processing (NIPS'17). 2017:6348-6358.
- [30] WU Y, SONG W, CAO Z, et al. Learning improvement heuristics for solving routing problems [J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33 (9): 5057-5069.
- [31] DACOSTA P, RHUGGENAATH J, ZHANG Y Q, et al. Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning [J]. arXiv:2004.01608, 2020.
- [32] HUANG L Y, CHEN X M, WEI H, et al. Branch and Bound in Mixed Integer Linear Programming Problems: A Survey of Techniques and Trends [J]. arXiv:2111.06257, 2021.
- [33] LAWLER E L, LENSTRA J K, RINNOOY K A H, et al. The Travelling Salesman: A Guided Tour of Combinatorial Optimization [J]. Journal of the Operational Research Society, 1986, 37(5):535-536.
- [34] LOMNICKI Z A. A Branch-and-Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem [J]. Journal of the Operations Research Society, 1965, 16:89-100.
- [35] CLAUSEN J, TRAFF J L. Implementation of parallel Branch-and-Bound algorithms experiences with the graph partitioning problem [J]. Annals of Operations Research, 1991, 33:331-349.
- [36] MAUTOR T, ROUCAIROL C. A new exact algorithm for the solution of quadratic assignment problems [J]. Discrete Applied Mathematics, 1994, 55(3):281-293.
- [37] LAND A H, DOIG A G. An automatic method for solving discrete programming problems [J]. Econometrica, 1960, 28(3):497-520.
- [38] CLAUSEN J. Branch and bound algorithms-principles and examples[J/OL]. https://janders.eecg.toronto.edu/1387/readings/b_and_b.pdf.
- [39] ACHTERBERG T, KOCH T, MARTIN A. Branching rules revisited[J]. Operations Research Letter, 2005, 33:42-54.
- [40] GAMRATH G. Improving Strong Branching by Domain Propagation[J]. EURO Journal on Computational Optimization, 2014, 2:99-122.
- [41] KILINÇ, MUSTAFA R L. Strong-branching inequalities for convex mixed integer nonlinear programs [J]. Computational Optimization and Applications, 2014, 59:639-665.
- [42] SANTANU S D, YATHARTH D, MARCO M, et al. A Theoretical and Computational Analysis of FullStrong-Branching[J]. arXiv:2110.10754, 2021.
- [43] MILLER C E, TUCKER A W, ZEMLIN R A. Integer programming formulation of traveling salesman problems [J]. Journal of the ACM, 1960, 7(4):326-329.
- [44] TOBIAS A, TIMO B. Hybrid branching [M] // Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Berlin: Springer, 2009:309-311.



WEI Niannan, born in 1998, bachelor, postgraduate. Her main research interests include logistics and supply chain management, and algorithm design.



HAN Shuguang, born in 1977, Ph. D, professor. His main research interests include logistics and supply chain management, and algorithm design.