

控制与决策

Control and Decision

基于数据关联感知的无监督深度融合指针网络模型

张长勇, 周虎

引用本文:

张长勇,周虎. 基于数据关联感知的无监督深度融合指针网络模型[J]. *控制与决策*, 2024, 39(2): 499–508.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.0925>

您可能感兴趣的其他文章

Articles you may be interested in

一种基于深度学习的时间序列预测方法

A time series prediction method based on deep learning

控制与决策. 2021, 36(3): 645–652 <https://doi.org/10.13195/j.kzyjc.2019.0809>

基于自注意力生成对抗网络的图像超分辨率重建

Image super-resolution reconstruction based on self-attention GAN

控制与决策. 2021, 36(6): 1324–1332 <https://doi.org/10.13195/j.kzyjc.2019.1290>

基于深度强化学习与迭代贪婪的流水车间调度优化

Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method

控制与决策. 2021, 36(11): 2609–2617 <https://doi.org/10.13195/j.kzyjc.2020.0608>

面向人机物三元数据的热轧调度问题研究

Research on hot rolling scheduling problem oriented to human-cyber-physical data

控制与决策. 2021, 36(11): 2825–2831 <https://doi.org/10.13195/j.kzyjc.2020.0551>

一种基于节点嵌入表示学习的社区搜索算法

Community search algorithm based on node embedding representation learning

控制与决策. 2021, 36(8): 1970–1976 <https://doi.org/10.13195/j.kzyjc.2019.1439>

基于数据关联感知的无监督深度融合指针网络模型

张长勇[†], 周 虎

(中国民航大学 电子信息与自动化学院, 天津 300300)

摘要: 为了提高组合优化问题可行解集合的收敛性和泛化性, 根据不同无监督学习策略的特点, 提出一种基于数据关联感知的深度融合指针网络模型 (DMAG-PN), 模型通过指针网络框架将 Mogrifier LSTM、多头注意力机制与图卷积神经网络三者融合. 首先, 编码器模块中的嵌入层对输入序列进行编码, 引入多头注意力机制获取编码矩阵中的特征信息; 然后构建数据关联模型探索序列节点间的关联性, 采用图卷积神经网络获取其多维度关联特征信息并融合互补, 旨在生成多个嵌入有效捕捉序列深层的节点特征和边缘特征; 最后, 基于多头注意力机制的解码器模块以节点嵌入数据和融合图嵌入数据作为输入, 生成选择下一个未访问节点的全局概率分布. 采用对称旅行商问题作为测试问题, 与当前先进算法进行对比, 实验结果表明, 所提出 DMAG-PN 模型在泛化性和求解精确性方面获得较大的改进与提高, 预训练好的 DMAG-PN 模型能够直接对大规模实例进行端到端的求解, 避免传统算法迭代搜索的过程, 具有较高的求解效率.

关键词: 指针网络; Mogrifier LSTM; 多头注意力机制; 图卷积神经网络; 旅行商问题; 数据关联

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0925

引用格式: 张长勇, 周虎. 基于数据关联感知的无监督深度融合指针网络模型 [J]. 控制与决策, 2024, 39(2): 499-508.

Data-correlation-aware unsupervised deep fusion pointer network model

ZHANG Chang-yong[†], ZHOU Hu

(College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China)

Abstract: In order to improve the convergence and generalization of feasible solution sets for combinatorial optimization problems, a data-correlation-aware deep fusion pointer network model (DMAG-PN) is proposed according to the characteristics of different unsupervised learning strategies. The model integrates Mogrifier LSTM, multi-head attention mechanism, and graph convolutional neural networks through a pointer network framework. Firstly, the embedding layer in the encoder module encodes the input sequence, and the multi-head attention mechanism is introduced to obtain the feature informations in the coding matrix. Secondly, the data correlation model is constructed to explore the correlation between the sequence nodes, and the graph convolution neural network is used to obtain the multi-dimensional correlation feature information, so as to generate multiple embeddings to effectively capture the deep node and edge features of the sequence. Finally, the decoder module based on multi-head attention mechanism takes node embedding data and fusion graph embedding data as inputs to generate a global probability distribution for selecting the next unvisited node. The symmetric traveling salesman problem is used as the test problem, and compared with the current advanced algorithms, the experimental results show that the proposed DMAG-PN model has been greatly improved in terms of generalization and accuracy. The pre-trained DMAG-PN model can directly solve large-scale instances, avoiding the iterative search process of traditional algorithms, and has a high solution efficiency.

Keywords: pointer network; Mogrifier LSTM; multi-head attention; graph convolutional neural networks; traveling salesman problem; data correlation

0 引言

组合优化问题 (Combinatorial optimization problem, COP) 广泛存在于国防、交通、航空航天、管

理决策等各个领域, 组合优化问题的求解一直是运筹学领域的研究热点之一^[1-3]. 从直观上看, 可以通过穷举排列获得问题最优解, 但随着问题规模增大, 很难

收稿日期: 2022-05-24; 录用日期: 2022-10-03.

基金项目: 国家自然科学基金项目 (62173331).

责任编委: 张国山.

[†]通讯作者. E-mail: cyzhang@cauc.edu.cn.

在多项式时间内获得问题最优解^[4]. 过去数十年中,许多学者提出大量的求解方法,从实现过程可分为两类:一类是基于全局最优的精确方法,代表算法有分支定界框架^[5]、动态规划法^[6]等;另一类是基于局部最优的近似方法,代表算法有局部搜索算法^[7-8]、禁忌搜索^[9]、模拟退火算法^[10]、蚁群优化算法^[11]等. 然而,根据“世界上没有免费的午餐”理论,没有一种求解算法是万能的. 其中:精确方法随着问题规模扩大,计算时间呈指数上升;启发式近似方法受算法背景影响,设计有效的启发式算法需要大量的专业知识以及试错实验,算法的泛化能力难以得到保证.

为了更好地达到“求解”与“泛化”能力之间的平衡,借助于深度神经网络强大的特征表示,国内外学者相继提出一批优秀的“端到端方法”^[12-19]. Vinyals等^[12]首次将神经网络融入序列映射模型用于旅行商问题的求解,提出经典指针网络模型,掀起了神经网络在组合优化领域的研究热潮. Bello等^[13]基于蒙特卡洛的策略梯度方法训练指针网络模型,提出了一种用于旅行商问题的神经网络框架. Deudon等^[14]将Transformer模型的自注意力机制与启发式算法相结合,提出了一种用于旅行商问题的组合优化框架. Kool等^[15]设计 rollout baseline 训练方法,基于Transformer模型中的注意力机制提出了一种用于路径问题的无监督模型. Zheng等^[16]基于Q-learning、Sarsa和MonteCarlo强化学习方法与Lin-Kernighan-Helsgaun算法相结合,提出了一种可变策略的VSR-LKH算法. Li^[17]等通过“分解”思想将多目标旅行商问题分解为标量子问题,基于指针网络提出了一种协同优化子参数的无监督模型. Ma等^[18]巧妙地将指针网络和图神经网络进行融合,提出了一种用于求解大规模旅行商问题的图指针网络. Gao等^[19]基于代表性模型和迁移学习策略,提出了一种用于旅行商问题的单模型多目标指针网络. 在上述算法中,尽管融入自注意力机制的指针网络模型更擅长捕捉单层面数据间的内部联系,但并不具备多头注意力捕捉多层面间稀疏化信息的优势,模型不能保持很好的鲁棒性. 此外,当前的研究往往聚焦于LSTM、多头注意力机制或者图神经网络在问题上的应用,很少有算法将两者或两者策略融合以优势互补地应用于组合优化中,提升模型的整体求解性能.

基于以上研究,本文基于Encoder-Decoder架构,将Mogrieff LSTM、多头注意力机制与图卷积神经网络三者融合,提出一种基于数据关联感知的无监督深度融合指针网络模型(deep multi-head attention and

graph neural pointer network, DMAG-PN). 针对输入序列样本构造数据关联模型,将序列优化问题转换为图优化问题;分阶段设计无监督优化策略,使DMAG-PN模型具备多策略融合互补的优势. 将DMAG-PN应用于对称旅行商问题^[20],实验结果与多个当前具有代表性的组合优化算法相比,所提出DMAG-PN算法的优化结果相比先进的深度学习算法提升约3%~5%,具有较强的泛化性与鲁棒性.

1 基于数据关联感知的深度融合指针网络模型

针对组合优化问题,提出一种基于数据关联感知的无监督深度融合指针网络模型(DMAG-PN),整体结构由数据关联感知图、双路融合编码器、点编码器和特征解码器4个模块组成,如图1所示. 首先编码器模块中的嵌入层对序列进行编码,引入多头注意力机制获取编码矩阵的特征信息;然后构建数据关联模型探索序列节点间的关联性,采用图卷积神经网络获取其多维度关联特征信息,并与多头注意力机制进行融合互补;接着采用点编码器对当前节点进行特征提取,以生成节点嵌入数据;最后解码器模块以节点嵌入数据和融合图嵌入数据作为输入,生成选择下一个未访问节点的全局概率分布. 以下对DMAG-PN关键模块的内部流程进行详细阐述.

1.1 基于深度融合机制的编码器模块

DMAG-PN编码器模块的主要功能是对组合优化问题的输入序列进行编码,将一维序列转换为高维度特征向量,充分探索序列中的结构信息,为下一步特征解码器的识别提供必要的信息. 如图1所示,所提出的DMAG-PN编码器主要由数据关联感知图、双路融合编码器、点编码器3个模块组成. 以下分别对各个模块进行详细阐述.

1.1.1 数据关联感知图

为了挖掘序列节点之间的潜在特征,显式地构建数据感知图关联来自序列空间中的相似节点. 给定组合优化的输入序列 $X = \{x_i | i = 1, 2, \dots, n\}$,采用KNN算法探索节点 x_i 序列空间中的 K 个近邻 $N_i = \{x_{ik} | k = 1, 2, \dots, K\}$. 如图2所示,通过在节点 x_i 与邻居节点 x_{ik} 之间分配无向边,构造无向图 $G = \{V, E\}$. 其中: $V = \{v_i = x_i, i = 1, \dots, n\}$ 为序列节点集, $E = \{e_{ij} = (x_i, x_{ik}) | x_{ik} \in N_i\}$ 为节点边缘集. 通过所构建的数据感知图,将序列优化问题转换为图优化问题,显式地探索图节点间的关联性,利用模型传播机制将图节点间的特征信息传递给图卷积神经网络,旨在进行多维度融合特征学习.

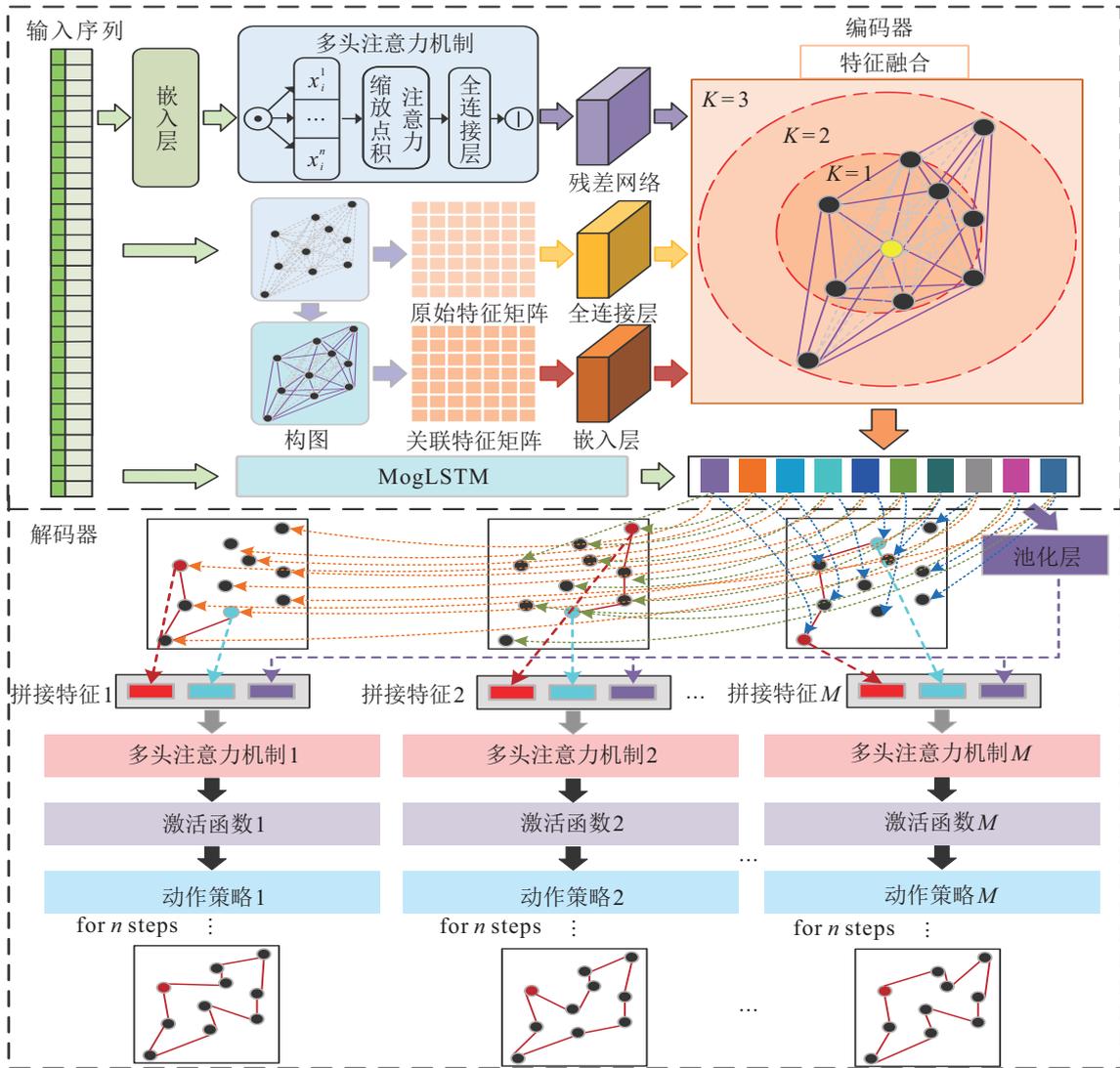


图1 DMAG-PN整体结构

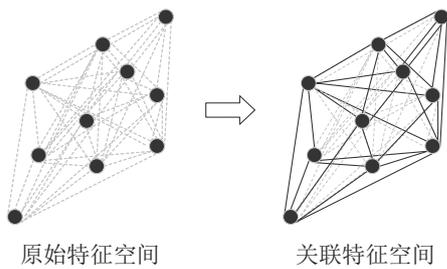


图2 数据关联感知图的构建

以无向对称旅行商问题为例,将节点城市坐标构造为无向图 $G = \{V, E\}$. 其中: $V = \{x_1, x_2, \dots, x_n\}$ 为包含所有城市坐标的节点集; $E = \{e_{11}, \dots, e_{nn}\}$ 为所有城市节点之间的连边集合,如果节点 x_i 与节点 x_j 相连接,则 $e_{ij} = 1$,否则 $e_{ij} = 0$.

算法1 数据关联图的构建.

输入: 输入序列 $X = \{x_i | i = 1, 2, \dots, n\}$,近邻的数量 k ,集合的大小 n ;

输出: 无向图 G .

函数: K-NN算法; dis: 距离计算函数,采用欧氏

距离; topK: 集合中最小距离索引选择函数.

step 1: 初始化 $G, G_{size} = [n, n]$

step 2: for i in rang (len S) do

step 3: $G_{ij} = 0$; //对角线元素为0

step 4: $e = knn(X[i], X, k)$; //寻找第 i 个节点的 k 近邻

step 5: $D = dis(X[i], e_i)$; //计算第 i 个节点与其他近邻的距离

step 6: $D = sort(D)$; //根据距离升序排序

step 7: $index = top K(D, n - 1)$;

step 8: for j in range(len(index)) do

step 9: $G_{ij} = dis(X[i], index[j])$; //构建图

step 10: end for

step 11: end for

1.1.2 双路融合编码器

基于深度融合机制的双路编码器由于融合操作的存在,在保持序列特征多样性方面有着很大的优越性. 双路融合编码器使用具备多头注意力机制和残

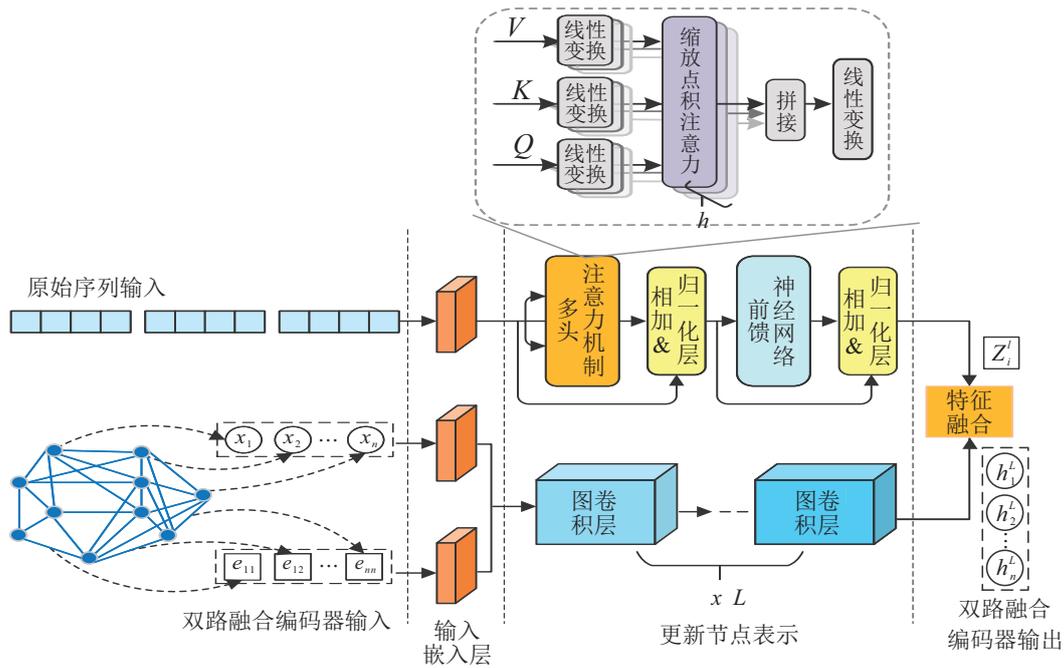


图3 双路融合编码器结构

差连接的Transformer编码器以及图卷积神经网络的图编码器,分别对原始序列样本和关联特征样本进行编码,具体结构如图3所示。

1) Transformer 编码器. 为对原始输入序列进行编码,Transformer 编码器嵌入层对其进行矩阵映射,采用多头注意力机制进行非线性的缩放点积运算,单个注意力层的计算方法如下:

$$X^l = H^l \in R^{(n+1) \times d}; \quad (1)$$

$$\text{Att}(Q, K, V) = \text{soft max} \left(\frac{Q^l K^{lT}}{\sqrt{d}} \right) V^l \in R^{(n+1) \times d}; \quad (2)$$

$$Q^l = X^l W_Q^L \in R^{(n+1) \times d}, W_Q^L \in R^{d \times d}; \quad (3)$$

$$K^l = X^l W_K^L \in R^{(n+1) \times d}, W_K^L \in R^{d \times d}; \quad (4)$$

$$V^l = X^l W_V^L \in R^{(n+1) \times d}, W_V^L \in R^{d \times d}. \quad (5)$$

其中: W_Q^L 、 W_K^L 、 W_V^L 为可学习的单个注意力层参数矩阵, X^l 为包含所有编码节点的矩阵, Q^l 、 K^l 和 V^l 分别为查询矩阵、键值矩阵和实值矩阵. 拼接 h 个单层注意力运算结果,得到多头注意力层的输出为

$$M\text{Att}(Q, K, V) = \text{Concat}(\text{Att}_1, \dots, \text{Att}_h) W^o, \quad (6)$$

其中 $W^o \in R^{(d \times h) \times d}$ 为多头注意力层参数矩阵. 多头注意力层的输出经过残差连接和正则化处理,旨在加速训练过程的收敛性,有

$$f = \text{Layer Norm}(X^l + M\text{Att}(Q, K, V)). \quad (7)$$

最后通过前向反馈网络,得到特征编码器的输出嵌入数据 Z_i^l 为

$$Z_i^l = \text{Layer Norm}(f + \text{FFN}(f)). \quad (8)$$

2) 图编码器. 基于1.1.1节的数据关联模型将输入序列转换为无向图 $G = \{V, E\}$, 为了对无向图 G 上的原始输入序列进行编码,图编码器对输入节点和邻居节点进行“原始嵌入”,将其映射到 h 维平面,采用多个图卷积层(graph convolutional layers)对 h 维的节点特征以及邻居节点特征进行聚合. 其中“原始嵌入”计算为

$$h_i^0 = A_0 x_i + b_0, \forall i \in \{1, 2, \dots, n\}; \quad (9)$$

$$e_{ij}^0 = A_1 e_{ij} + b_1, \forall j \in \{1, 2, \dots, n\}. \quad (10)$$

其中: h_i^0 为原始节点表示, e_{ij}^0 为原始邻居节点表示, $A_0 \in R^{h \times 2}$ 和 $A_1 \in R^h$ 为可学习的权重参数矩阵, x_i 为节点坐标, e_{ij} 为边缘节点坐标, $b_0 \in R^h$ 和 $b_1 \in R^h$ 为偏置参数.

采用多个图卷积层的方式可以有效捕获节点表示间的关联性特征,给定第 l 层图卷积网络的输入 X^{l-1} ,第 l 层图卷积网络的输出 X^l 可以表示为

$$X^l = \gamma X^{l-1} W_g + (1 - \gamma) \varphi_\theta \left(\frac{X^{l-1}}{|N(i)|} \right). \quad (11)$$

其中: $X^l \in R^{N \times d_l}$ 为包含节点表示 $(h_i^0, h_i^1, \dots, h_i^N)$ 的节点集, $\varphi_\theta : R^{N \times d_{l-1}} \rightarrow R^{N \times d_l}$ 为聚合函数, γ 为可训练参数, $W_g \in R^{d_{l-1} \times d_l}$ 为可训练的权重矩阵, $N(i)$ 为包含邻居节点表示 $(e_{ij}^0, e_{ij}^1, \dots, e_{ij}^N)$ 的节点集. 最后 L 层图卷积神经网络的输出即为更新后的 h 维节点表示 h_i^L .

在获得嵌入数据 Z^l 和 h 维节点表示 h_i^L 的基础

上,通过图融合模块将异构元数据的嵌入融合到一个共享的潜在图空间,通过一个全连接层和softmax函数以获得最终的融合图嵌入 $Z_i^f \in R^{N \times D}$,有

$$Z_i^f = \text{soft max}(\text{fusion}(Z_i^l, h_i^l)) = \text{soft max}(\vartheta_\theta(\text{Add}(Z_i^l, h_i^l))). \quad (12)$$

其中: ϑ_θ 为聚合函数, $\text{Add}()$ 为矩阵元素相加运算.

1.1.3 点编码器

基于状态交互机制的点编码器由于MogLSTM神经网络的存在,在探索序列潜在特征的分布性方面有着很大的优越性. 将当前节点表示 x_i 映射得到 h 维向量 $x_i^l \in R^d$,采用MogLSTM神经网络对 h 维向量 x_i^l 进行编码,得到最终的输出节点嵌入数据 H_i^l . 其中MogLSTM由遗忘门 F_t 、输入门 I_t 、输出门 O_t 和记忆细胞 c 构成,数学描述为

$$\text{MogLSTM}(x_i, c_{t-1}, h_{\text{prev}}) = (c, h). \quad (13)$$

各个门的计算方法如下所示:

$$F_t = \sigma(W_{fx}X + W_{fh}h_{\text{prev}} + b_f); \quad (14)$$

$$I_t = \sigma(W_{ix}X + W_{ih}h_{\text{prev}} + b_i); \quad (15)$$

$$\widetilde{C}_t = \tanh(W_{tx}X + W_{th}h_{\text{prev}} + b_t); \quad (16)$$

$$O_t = \sigma(W_{ox}X + W_{oh}h_{\text{prev}} + b_o); \quad (17)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \widetilde{C}_t; \quad (18)$$

$$H_i^l = H = O_t \odot \tanh(C_t); \quad (19)$$

$$x^j = 2\sigma(Q^j h_{\text{prev}}^{j-1}) \odot x^{j-2}, j \in [1, 2, \dots, r]; \quad (20)$$

$$h_{\text{prev}}^j = 2\sigma(R^j x^{j-1}) \odot h_{\text{prev}}^{j-2}, j \in [1, 2, \dots, r]. \quad (21)$$

其中: σ 为sigmoid函数, \odot 为向量元素积符号, W_* 为权重参数矩阵, b_* 为偏置参数, i 为输入 x 和隐藏状态 h_{prev} 的交互轮数.

点编码器的输出嵌入数据 H_i^l 即为MogLSTM神

经网络新一轮隐藏状态 H . MogLSTM的核心在于引入额外的门控运算,使得输入 x 、状态 h_{prev} 在进入LSTM之前进行多轮交互计算,以增强点编码器的上下文信息提取能力,为当前时间戳解码器的特征提取和下一时间戳点编码器的数据编码提供必要的嵌入数据.

1.2 基于注意力机制的解码器模块

DMAG-PN解码器模块的主要功能是对编码器中获得的高维度特征向量进行重构,依据当前节点表示 H_i^l 和融合图嵌入 Z_i^f ,采用多头注意力层为下一步选择合适的节点,得出最优解序列.

如图4所示,所提出的DMAG-PN解码器由 M 个结构相同、参数独立的解码器组成,为了对嵌入数据 H_i^l 、 Z_i^f 进行解码,以 $m \in \{1, 2, \dots, M\}$ 索引的方式构造上下文向量 c_t^m ,有

$$c_t^m = \begin{cases} \text{concat}(\overline{Z}, H_{\pi_1^m}^l, v_1^m), & t = 1; \\ \text{concat}(\overline{Z}, H_{\pi_{t-1}^m}^l, Z_{\pi_{t-1}^m}^f), & t > 1. \end{cases} \quad (22)$$

其中: $\text{concat}()$ 为拼接运算符, $\overline{Z} = \frac{1}{n} \sum_{i=1}^n Z_i^f$ 为融合图嵌入 Z_i^f 的平均值, $H_{\pi_{t-1}^m}^l$ 、 $Z_{\pi_{t-1}^m}^f$ 分别为解码器选择的当前节点嵌入数据和第1个节点嵌入数据, v_1^m 为 $t = 1$ 时的输入占位符. 将上下文向量 c_t^m 、节点表示 $(h_i^0, h_i^1, \dots, h_i^N)$ 输入到基于指针网络的注意力层,以得到指针向量 u_i^j ,有

$$u_i^j = \begin{cases} v^T \tanh(W_r r_j + W_q q), & j \neq \pi_k^m, \forall k < j; \\ -\infty, & \text{otherwise.} \end{cases} \quad (23)$$

其中: u_i^j 为向量 u_i 的第 j 项元素, W_r 、 W_q 为可训练参数矩阵, $q = H_i^l$ 为MogLSTM中隐藏状态的查询向量, $r_j = Z_i^f$ 为包含图融合模块所有节点信息的参考

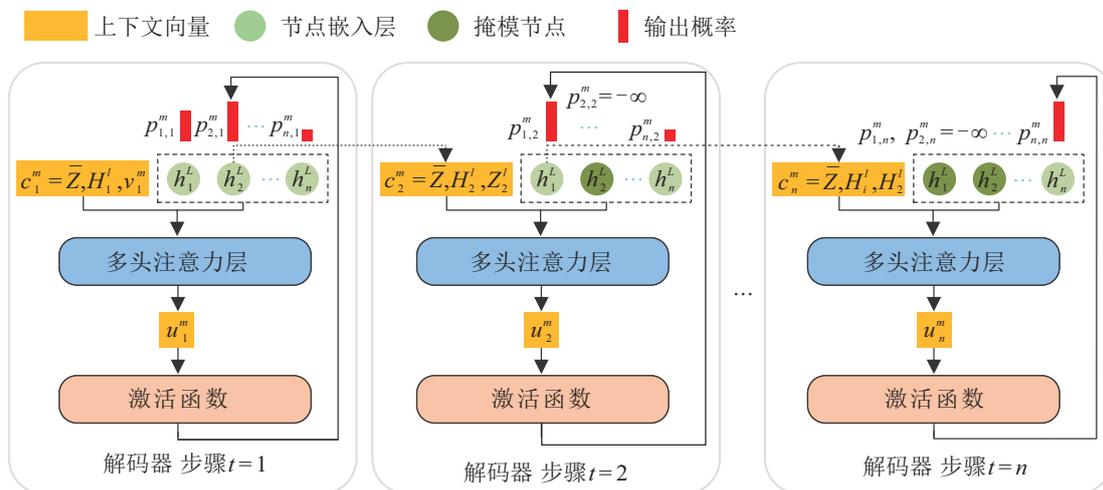


图4 Decoder解码过程示意图

向量. 最后, 通过 softmax 函数生成所有候选节点的概率分布, 以此选择下一个有效节点

$$p_{i,t}^m = p_\theta(\pi_t^m = i | x, \pi_{1:t-1}^m) = \text{soft max}(u_i^j). \quad (24)$$

1.3 训练方法

对于随机输入实例 s , 每个编码器生成一个排列 π^m , 以最佳排列 π_t^m 对应的模型作为基线策略. 为训练模型, 以期望成本总和 $R(\pi^m)$ 的方式定义模型的损失函数 $\text{loss}(\theta|s)$, 有

$$b(s) = \min_m R(\pi_1^m, \dots, \pi_n^m | s), \quad (25)$$

$$\pi_t^m = \arg \max_i (p_\theta(\pi_t^m = i | s, \pi_{1:t-1}^m)), \quad (26)$$

$$\text{loss}(\theta|s) = \sum_m E_{p_\theta(\pi|s)} [R(\pi^m)]. \quad (27)$$

使用带确定性贪心基线值 (greedy rollout baseline) 的强化学习算法对参数 θ 进行更新, 不断地最小化策略梯度, 即

$$\begin{aligned} \nabla_\theta \text{loss}(\theta|s) = \\ \sum_m E_{p_\theta(\pi|s)} \times [(R(\pi^m) - b(s)) \nabla_\theta \log p_\theta(\pi^m|s)]. \end{aligned} \quad (28)$$

训练算法的伪代码如下.

算法2 REINFORCE with Rollout Baseline.

输入: 轮数 E , 每轮步数 T , 批量尺寸 B , 显著性水平 α , 演员网络未训练参数 θ , 策略基准网络未训练参数 θ^b ;

输出: 训练参数集 θ .

step 1: Init θ, θ^b ;

step 2: for epoch = 1, 2, ..., E do

step 3: for step = 1, 2, ..., T do

step 4: $S_i \leftarrow$ 随机生成实例, $\forall i \in \{1, 2, \dots, B\}$;

step 5: $\{\pi_i^1, \dots, \pi_i^M\} \leftarrow p_\theta(\pi_i^m | s_i, \pi_{1:t-1}^m), \forall i \in \{1, 2, \dots, M\}, m \in \{1, 2, \dots, M\}$;

step 6: $\{\pi_i^{b,1}, \dots, \pi_i^{b,M}\} \leftarrow p_{\theta^b}(\pi_i^m | s_i, \pi_{1:t-1}^m), \forall i \in \{1, 2, \dots, M\}, m \in \{1, 2, \dots, M\}$;

step 7: $b(s) = \min(R(\pi_i^{b,m})), \forall i \in \{1, 2, \dots, M\}, m \in \{1, 2, \dots, M\}$;

step 8: $\nabla \text{loss} \leftarrow \sum_{i=1}^B \sum_{m=1}^M (R(\pi_i^m) - b(s)) \nabla \log p_\theta(\pi_i^m)$;

step 9: $\theta \leftarrow \text{Adam}(\theta, \nabla \text{loss})$;

step 10: end for

step 11: if Paired $t - \text{test}(p_\theta, p_{\theta^b}) < \alpha$ then

step 12: $\theta^b \leftarrow \theta$

step 13: end if

step 14: end for

1.4 与现有方法的比较

本节主要讨论 DMAG-PN 与相关方法之间的差异. 以对称旅行商问题为测试例, 选取 2 种精确算法求解器: Concorde、LKH3^[7], 4 种经典近似算法: Nearest Neighbor、2-Opt、Farthest Insertion、OR-Tools^[21], 以及近年 4 种深度学习代表性算法: Pointer Net^[12]、Attention Model^[15]、Graph Pointer Networks^[18]、s2v-DQN^[22].

Concorde、LKH3 求解器: 公认较好的最优解和次优解求解器, 将原始问题分解为子问题, 迭代改进获得更好的较优解. 求解器对小规模数据样本能在可接受时间内获得最优结果; 缺陷在于大规模数据样本, 难以在“适当”时间内获得较为理想的结果.

NN、FI、2-Opt、OR-Tools 近似算法: 该类近似算法通过将问题切分成最大独立子集问题, 能够在可接受时间内构造出较好的可行解, 但受算法机制的影响一定程度上容易陷入局部最优.

Pointer Net: Ptr-Net 的核心是序列到序列的映射过程, 结合 Attention 机制逐步构造可行解, 对于小规模数据样本, 能够快速得到接近最优解的效果; 对于大规模数据样本, 其与最优解还存在一定差距.

Attention Model: 效仿人类生存机制, 捕捉重要信息的权重分配, 极大地提高模型的效率和准确性. 但是注意力模型无法学习数据空间中的顺序关系, 在大规模数据样本中, 与最优解依然存在一定差距.

s2v-DQN: 从空间结构相似性的角度定义顶点相似度, 捕捉除近邻顶点以外的结构特征, 能够在短时间内生成高质量解, 但在解的过程中需要人为设计特定于某个问题的辅助函数以更新神经网络.

Graph Pointer Networks: GPN 使用图嵌入层扩展指针网络, 以加快模型收敛速度, 在大规模数据样本上获得了显著优势, 而在小规模数据上效果不及 Attention Model 等深度学习模型.

综合以上分析, 传统 Concorde、LKH3 求解器在大规模数据样本上综合效果表现不佳; 近似算法可以处理大规模数据样本, 但容易陷入局部最优; 相关深度学习方法求解效率高、泛化能力强, 但在某些方面也存在着不足. 本文提出的 DMAG-PN 模型深入分析影响算法性能的各种因素, 深度融合多种优化策略, 优势互补地提高算法综合性能. 设计如下实验以佐证上述分析结论.

2 实验仿真与结果分析

对称旅行商问题是一类经典的组合优化问题, 给定 n 个城市节点坐标 $\{x_1, x_2, \dots, x_n\} \in R^2$, 商人按照一定顺序从某个城市出发, 寻找遍历所有城市仅一次的最佳排列 $\pi = (\pi_1, \pi_2, \dots, \pi_n)$, 使得商人走的 Hamilton 路径最短, 即

$$L(\pi, V) = \|x_{\pi_1} - x_{\pi_n}\|_2 + \sum_{i=1}^{n-1} \|x_{\pi_i} - x_{\pi_{i+1}}\|_2. \quad (29)$$

其中: $V = \{x_1, x_2, \dots, x_n\}$ 为城市节点集; $\pi_i \in \{1, 2, \dots, n\}$, 对于任意的 $i \neq j$ 均有 $\pi_i \neq \pi_j$. 对于对称旅行商问题, $d(x_{\pi_i}, x_{\pi_j})$ 表示城市 x_{π_i} 与城市 x_{π_j} 之间的欧氏距离, 有

$$d(x_{\sigma(i)}, x_{\sigma(j)}) = \|x_{\sigma(i)} - x_{\sigma(j)}\|_2, \quad (30)$$

其中 $d(x_{\pi_i}, x_{\pi_j}) = d(x_{\pi_j}, x_{\pi_i}), \forall i, j \in \{1, 2, \dots, n\}$.

2.1 实验环境和数据集

在对称旅行商问题中验证所提出 DMAG-PN 的性能. 实验基于 Linux 操作系统, 使用 Python 语言, 在 Pytorch 深度学习框架下进行模型搭建. 平台主要硬件参数为: Intel Xeon E5-2690 v4, 512 G 内存, NVIDIA Tesla P100, 16 G 显存.

实验采用从均匀分布 $x \sim U(0, 1)$ 中随机采样的方式生成旅行商节点坐标, 数据集有: 1) 具备 50 个节点的 TSP50 实例; 2) 具备 100 个节点的 TSP100 实例. 为了验证 DMAG-P 模型在不同规模 TSP 实例上的性能, 分别设计两组实验: 1) 小规模 TSP 实验; 2) 大规模 TSP 实验.

2.2 评价指标

与文献 [15, 18] 等本领域相关成果一样, 本文主要采用平均路径总长度 (Tour Len)、最优差距 (Gap)、评估时间 (Time) 等典型指标评估 DMAG-PN 模型的性能. 具体定义如下.

Tour Len: 体现了商人遍历全部城市的平均路径总长度, 计算方法如下:

$$\bar{R} = \frac{1}{n} \sum_{i=1}^n R_i = L(\pi, V). \quad (31)$$

Gap: 体现了 DMAG-PN 模型与最优求解器 Concorde 之间的平均最优差距, 计算方法如下:

$$\text{Gap} = \frac{1}{n} \sum_{i=1}^n \left(\frac{R'_i}{R_i} - 1 \right). \quad (32)$$

其中: n 为实例数, R_i 为 DMAG-PN 模型的最佳结果, R'_i 为 Concorde 求解器提供的最优解.

Time: 体现了 DMAG-PN 模型生成 10 000 个解的

平均时间, 与文献 [18] 保持一致, 模型的运行设备均为 Nvidia Tesla P100 GPU.

2.3 实验参数

为了使 DMAG-PN 模型达到“求解质量”与“求解时间”之间的平衡, 其基本运行参数设定如表 1 所示.

表 1 基本运行参数设置

参数	值	参数	值
Graph Embedding	3	Optimizer	Adam
Transformer Encoder	6	Learning rate	1e-4
Epoch	100	Learning rate decay	0.96
Batch size	512	Interact parameter	6
Training steps	2 500		

2.4 实验结果与分析

2.4.1 小规模实验

为了验证 DMAG-PN 在小规模旅行商问题上的性能, 选取 TSP50 实例训练模型, 将 DMAG-PN 在小规模 TSP 问题上的性能与现有算法进行比较.

实验结果如表 2 所示, DMAG-PN 在 TSP50 实例上的平均路径总长度为 5.708、时间为 2.31 s, 效果远远优于启发式近似算法、深度学习算法^[12-15, 18], 与最优求解器 Concorde 之间的差距仅有 0.14%, 其中带 * 的结果均出于文献 [18]. 因此所提出的 DMAG-PN 模型具有更好的求解性能, 这主要是由于 DMAG-PN 在保持序列特征多样性、增强搜索能力方面有着很大的优势. 由此表明, 基于 Transformer 和图神经网络的融合编码, 辅以状态交互的点编码策略, 以及基于多头注意力的解码策略等有助于提高算法高维度特征多样性, 提升全局搜索性能和解集的收敛性.

表 2 不同算法在 TSP50 实例上的比较

算法	TSP50		
	Tour Len	Gap / %	Time / s
LKH3*	5.700	0.00	300
Concorde*	5.700	0.00	120
Nearest Neighbor	6.934	22.65	4.24
2-Opt	6.117	7.32	9.74
Farthest Insertions	6.010	5.44	5.44
OR-Tools	5.723	0.40	50.03
Pointer Net*	7.660	34.39	-
s2v-DQN*	5.950	4.39	-
Attention Model*	5.800	1.75	2
GPN*	5.959	4.54	1.75
GPN+2Opt*	5.867	2.93	6.5
Ours	5.708	0.14	2.31

表3 不同算法在大规模旅行商问题上的比较

算法	TSP250			TSP500			TSP750			TSP1000		
	Tour Len	Gap/%	Time/s									
Concorde*	11.89	0.00	1 894	16.55	0.00	13 902	20.10	0.00	32 993	23.11	0.00	47 804
LKH*	11.893	0.03	9 792	16.542	-0.05	23 070	20.129	0.14	36 840	23.130	0.09	50 680
Nearest Neighbor	14.942	25.67	21	20.794	25.64	60	25.246	25.60	104	28.958	25.31	156
2-Opt	13.263	11.55	356	18.612	12.46	1 406	22.672	12.80	3 409	26.136	13.09	6345
Farthest Insertion	13.024	9.54	50	18.3	10.57	199	22.318	11.03	520	25.766	11.49	1 056
OR-Tools*	12.652	6.41	5 000	17.653	6.66	5 000	22.933	14.09	5 000	28.332	22.60	5 000
Pointer Net*	14.249	19.84	29	21.409	29.36	280	27.382	36.23	782	32.714	41.56	3 133
Attention Model*	14.032	18.02	2	24.789	49.78	14	28.281	40.70	42	34.055	47.36	136
s2v-DQN*	13.079	10.00	476	18.428	11.35	1 508	22.550	12.19	3 182	26.046	12.70	5 600
GPN*	13.679	15.05	32	19.605	18.46	111	24.337	21.08	232	28.471	23.20	393
GPN+2Opt*	12.942	8.85	214	18.358	10.92	974	22.541	12.14	2 278	26.129	13.06	4 410
Ours	13.397	12.67	20	18.964	14.59	49	23.259	15.72	96	26.894	16.37	-
Ours+2Opt	12.846	8.04	235	18.013	8.44	1 146	21.959	9.25	2 409	25.278	9.38	-

2.4.2 大规模实验

为了验证DMAG-PN在大规模旅行商问题上的性能,实验选取TSP100实例训练模型,并将其泛化到更大规模旅行商问题.与经典近似算法、当前效果较好的深度学习算法进行比较,实验结果如表3所示. DMAG-PN在TSP250、TSP500、TSP750和TSP1000实例上的平均路径总长度分别为13.397、18.964、23.259、26.894,最优差距分别为12.67%、14.59%、15.72%、16.37%,其中带*的结果均出于文献[18].

对比近似算法(Nearest Neighbor、2-Opt、Farthest Insertions和OR-Tools),在大规模数据样本上,DMAG-PN的平均路径总长度与近似算法相当;由于受到数据集规模的影响,近似算法的平均时间指标比DMAG-PN均高出65%以上.对比深度学习算法(Pointer Net、Attention Model、s2v-DQN和GPN),DMAG-PN的各项指标比其中综合效果最好的GPN均优于3%~5%.因此,在大规模样本泛化问题上,深度学习方法在“求解质量”与“求解时间”之间达到了平衡,整体效果优于近似算法.本文提出的DMAG-PN算法更是表现出优异的均衡性和泛化性,这主要是由于DMAG-PN在探索序列样本间的潜在关联性、协同编码器多策略融合的稳定性以及增强解码器特征向量的重构性等方面有着很大的优势.由此表明,所提出的DMAG-PN模型在充分发挥深度学习优势的基础上,能够有效感知序列间的潜在关系、协同多路编码/解码的融合策略以更高效地训练指针网络模型,解决其他深度学习模型由于数据的相似性、数据

量小而造成的训练效率低、泛化性能差等问题,从而验证DMAG-PN模型的有效性和泛化优势.

GPN模型在初始解的基础上使用2OPT搜索策略,为了与之全面比较,DMAG-PN亦辅以2OPT策略提高模型求解精度,平均路径总长度均提高2.5%以上,时间指标不如前者,其中TSP1000实例受限于显存限制,无法以相同参数进行时间指标对比. DMAG-PN在大规模旅行商问题上的可视化如图5所示.

2.4.3 参数敏感性实验

本节实验测试编码过程中MogLSTM的交互轮数 i 对DMAG-PN模型的影响.实验将 i 的取值区间设定在[3, 10],分别在TSP250、TSP500、TSP750和TSP1000实例上每组独立运行10次并取平均值,实验结果如图6所示.

综合以上实验结果可见,随着交互参数 i 的增大,平均路径总长度总体呈下降趋势,DMAG-PN模型的最优结果多集中于预训练模型参数 $i=6$ 的附近.交互参数 i 的小范围变化并没有对DMAG-PN性能产生较大影响,即DMAG-PN对参数 i 不敏感.

2.4.4 特征融合实验

双路融合编码器设计中,式(12)采用相加方式(ADD)进行特征融合,此外,融合方式还包括级联(CONCAT)、最大值(MAX)和平均值(AVERAGE).本节实验以TSP50为测试例,对4种特征融合方式进行对比,参数设置同2.3节.实验结果如表4所示,ADD的最优差距指标分别优于其他3种融合方式0.56%、0.19%、0.49%,平均时间基本保持一致.

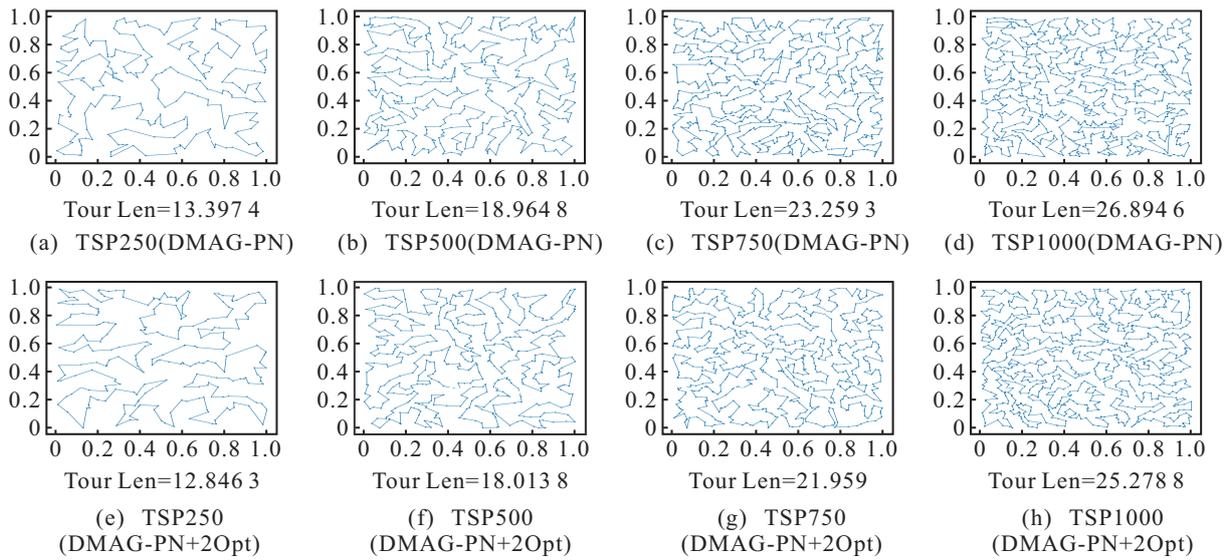


图5 DMAG-PN在大规模旅行商问题上的示例

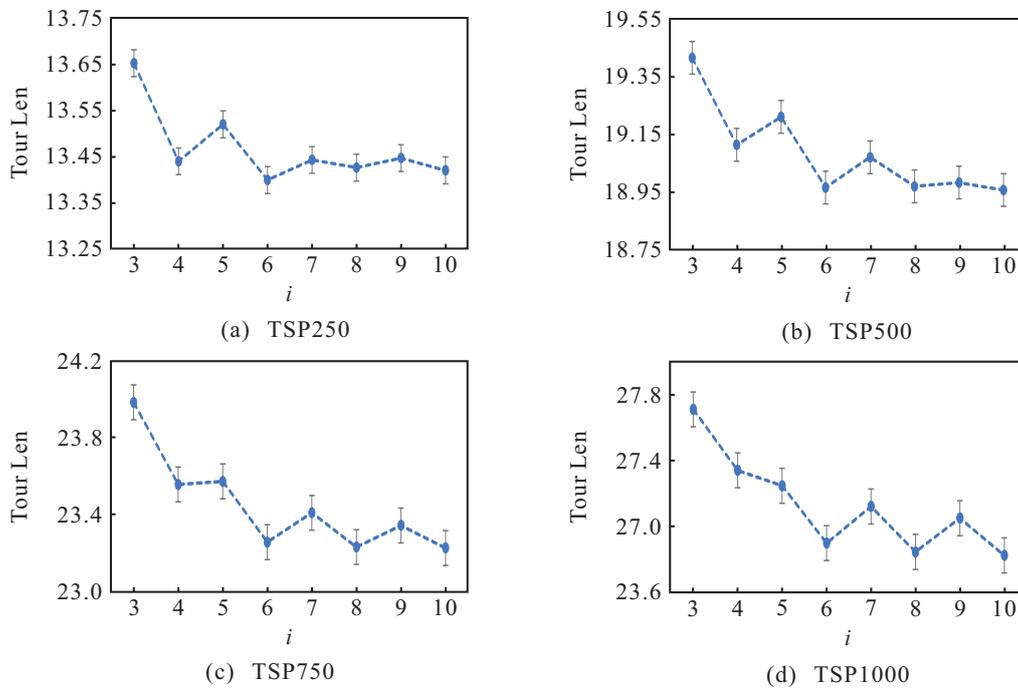


图6 DMAG-PN模型参数敏感性实验

表4 DMAG-PN特征融合实验

模型	Tour Len	Gap/%	Time/s
DMAG-PN (ADD)	5.708	0.14	2.31
DMAG-PN (MAX)	5.740	0.70	2.43
DMAG-PN (AVERAGE)	5.719	0.33	2.31
DMAG-PN (CONCAT)	5.736	0.63	2.36

3 结论

针对当前组合优化算法收敛性和泛化性差的现状,本文提出一种基于数据关联感知的无监督深度融合指针网络模型.由数据感知图关联序列样本间的隐藏信息,双路编码器提取并融合序列原始特征和关联特征,点编码器提取当前序列特征,通过掩模解码

器网络计算序列样本的概率分布,完成整个序列的组合优化.以对称旅行商问题为测试例进行大量实验,将所提出算法与经典近似算法以及4种先进深度学习算法进行对比,实验结果表明深度融合机制可显著提升模型的泛化性和求解精确性.在未来的研究中,将继续探索优化策略之间的融合关系,进一步扩展模型,以解决更复杂的组合优化问题,如离线三维装箱问题、在线三维装箱问题等.

参考文献(References)

[1] Papadimitriou C H, Steiglitz K. Combinatorial optimization: Algorithms and complexity[M]. New York: Courier Corporation, 1998.

- [2] Wang Q, Tang C L. Deep reinforcement learning for transportation network combinatorial optimization: A survey[J]. Knowledge-Based Systems, 2021, 233: 107526.
- [3] Mazyavkina N, Sviridov S, Ivanov S, et al. Reinforcement learning for combinatorial optimization: A survey[J]. Computers & Operations Research, 2021, 134: 105400.
- [4] Festa P. A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems[C]. The 16th International Conference on Transparent Optical Networks. Graz, 2014: 1-20.
- [5] Kinable J, Smeulders B, Delcour E, et al. Exact algorithms for the equitable traveling salesman problem[J]. European Journal of Operational Research, 2017, 261(2): 475-485.
- [6] 胡理嫚, 李志伟, 刘雪垠, 等. 基于动态规划的装配线物料搬运节能调度方法[J]. 系统工程理论与实践, 2018, 38(9): 2424-2433.
(Hu L M, Li Z W, Liu X Y, et al. Dynamic programming-based energy-saving scheduling for material handling in assembly lines[J]. Systems Engineering—Theory & Practice, 2018, 38(9): 2424-2433.)
- [7] Helsgaun K. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems[J]. Journal of Roskilde University, 2017, 126(1): 24-50.
- [8] 栗三一, 李文静, 乔俊飞. 一种基于密度的局部搜索 NSGA2 算法[J]. 控制与决策, 2018, 33(1): 60-66.
(Li S Y, Li W J, Qiao J F. A local search strategy based on density for NSGA2 algorithm[J]. Control and Decision, 2018, 33(1): 60-66.)
- [9] 李国明, 李军华. 基于混合禁忌搜索算法的随机车辆路径问题[J]. 控制与决策, 2021, 36(9): 2161-2169.
(Li G M, Li J H. Stochastic vehicle routing problem based on hybrid tabu search algorithm[J]. Control and Decision, 2021, 36(9): 2161-2169.)
- [10] 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.)
- [11] Wang Y, Han Z P. Ant colony optimization for traveling salesman problem based on parameters optimization[J]. Applied Soft Computing, 2021, 107: 107439.
- [12] Vinyals O, Fortunato M, Jaitly N. Pointer networks[J]. Advances in Neural Information Processing Systems, 2015, 2015-January: 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] Deudon M, Cournut P, Lacoste A, et al. Learning heuristics for the TSP by policy gradient[C]. Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Cham: Springer International Publishing, 2018: 170-181.
- [15] Kool W, van Hoof H, Welling M. Attention, learn to solve routing problems![J/OL]. 2018, arXiv: 1803.08475.
- [16] Zheng J Z, He K, Zhou J R, et al. Combining reinforcement learning with Lin-Kernighan-Helsgaun algorithm for the traveling salesman problem[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(14): 12445-12452.
- [17] Li K W, Zhang T, Wang R. Deep reinforcement learning for multiobjective optimization[J]. IEEE Transactions on Cybernetics, 2021, 51(6): 3103-3114.
- [18] Ma Q, Ge S W, He D Y, et al. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning[J/OL]. 2019, arXiv: 1911.04936.
- [19] Gao L Y, Wang R, Liu C, et al. Multi-objective pointer network for combinatorial optimization[J/OL]. 2022, arXiv: 2204.11860.
- [20] Cinar A C, Korkmaz S, Kiran M S. A discrete tree-seed algorithm for solving symmetric traveling salesman problem[J]. International Journal Engineering Science and Technology, 2020, 23(4): 879-890.
- [21] Perron L, Furnon V. Google's OR-tools[DB/OL]. [2022-05-21]. <https://developers.google.com/optimization>.
- [22] Khalil E, Dai H, Zhang Y Y, et al. Learning combinatorial optimization algorithms over graphs[J]. Advances in Neural Information Processing Systems, 2017, 30(3): 1-11.

作者简介

张长勇(1978—), 男, 副教授, 博士, 从事机场智能与自动化、智能优化理论等研究, E-mail: cyzhang@cauc.edu.cn;

周虎(1997—), 男, 硕士生, 从事机场智能与强化学习组合优化算法的研究, E-mail: zhou097@163.com.