

基于 Graph Transformer 的无人机全覆盖路径规划方法

陈 旭¹, 王从庆¹, 曾 强², 李 战²

(1. 南京航空航天大学 自动化学院, 南京 211106;

2. 中航贵州飞机有限责任公司, 贵州 安顺 561000)

摘要: 为了实现无人机对三维结构的损伤检测, 同时避免无人机与三维结构之间的碰撞, 保证检测过程的准确、高效, 针对无人机全覆盖路径规划问题, 提出了一种基于 Graph Transformer 的无人机全覆盖路径规划方法: 将其视为旅行商问题的变体, 在全连接图上用图神经网络进行求解; 在图神经网络中引入了注意力模块, 缓解了图神经网络中稀疏消息传递的局限性; 结合图卷积和注意力机制对节点和边进行特征提取; 在解码器中评估每条边在解中存在的概率, 生成概率热力图; 通过波束搜索获得初步解, 并使用局部搜索进行优化; 实验结果表明, 与基于强化学习、搜索的深度学习方法以及改进的蚁群方法和遗传算法相比, 该方法在性能表现、泛化性等方面具有显著优势; 并适用于二维和三维空间中的欧氏距离及非欧氏距离情况, 在无人机导航和全覆盖路径规划方面具有很好的应用价值。

关键词: Graph Transformer; 全覆盖路径规划; 图神经网络; 旅行商问题; 注意力机制

A Full-Coverage Path Planning Method for UAVs Based on Graph Transformer

CHEN Xu¹, WANG Congqing¹, ZENG Qiang², LI Zhan²

(1. School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;

2. China National Guizhou Aviation Industry (Group) Co., Ltd., Anshun 561000, China)

Abstract: To achieve the damage detection of three-dimensional structures by UAVs while avoiding collisions between UAVs and three-dimensional structures, and ensuring the accuracy and efficiency during inspection process, this paper proposes a full-coverage path planning method for UAVs based on GraphTransformer. The problem is treated as a variant of the traveling salesman problem and is solved using a graph neural network (GNN) on a fully connected graph. An attention module is introduced in the GNN to alleviate the limitation of sparse message passing in the network. The method combines a graph convolution and attention mechanism to extract the features from their nodes and edges. In the decoder, the existing probability of each edge in the solution is evaluated to generate a probability heat map. An initial solution is obtained using beam search, which is then optimized through local search. Experimental results show that, compared with the reinforcement learning, search deep learning, improved ant colony optimization, and genetic algorithms, the proposed method has significant advantages of performance and generalization, which is suitable for both Euclidean and non-Euclidean distances in two-dimensional and three-dimensional spaces, demonstrating a great potential in UAV navigation and full-coverage path planning.

Keywords: graph transformer; full-coverage path planning; graph neural networks; traveling salesman problem; attention mechanism

0 引言

无人机凭借着高效、灵活的特点, 在多个领域尤其是在三维结构的视觉检测任务中得到了广泛的应用。这些任务通常包括视图规划和路径规划两个关键环节, 其

中路径规划解决的是在避免与待检测结构发生碰撞的同时, 如何有效地遍历所有预定视点的问题, 也就是旅行商问题^[1] (TSP, traveling salesman problem)。

旅行商问题是一个经典的组合优化问题, 广泛应用

收稿日期:2024-11-12; 修回日期:2024-12-21。

作者简介:陈 旭(1999-), 男, 硕士研究生。

引用格式:陈 旭, 王从庆, 曾 强, 等. 基于 Graph Transformer 的无人机全覆盖路径规划方法[J]. 计算机测量与控制, 2025,

33(12):224-229,277.

于计算机科学和运筹学领域^[2]。其核心目标是在一张由节点和边构成的图中, 找到一条遍历所有节点并回到起点的路径, 使得路径上所有边的权重和最小化。TSP 不仅在理论研究中具有重要意义, 而且在物流、制造、运输等实际应用中也发挥着关键作用^[3]。

由于旅行商问题的 NP-Hard 特性, 不能在多项式时间内求得解, 所以很难得到旅行商问题的最优解。这导致传统的解决方案不仅需要长时间的计算, 而且大多依赖于专业知识和手工设计的规则。传统的 TSP 求解方法可以分为精确求解和启发式求解两种。精确求解方法如 Concorde 算法, 利用分支界定和线性规划技术, 能在多项式时间内找到较好的解, 然后不断优化以求得最优解。另一方面, 启发式求解算法如 Lin-Kernighan-Helsgaun (LKH) 求解器^[4], 通过局部优化和路径交换策略不断改进解决方案。虽然这些传统方法能够解决 TSP 问题, 但它们的设计复杂, 计算时间长, 难以满足即时的应用需求。

近几年, 随着深度学习的火热, 通过深度学习求解旅行商问题得到了很多的专家和学者的关注。最早使用神经网络求解旅行商问题的工作包括 Hopfield 神经网络^[5]和自组织神经网络^[6], 这些方法相对简单, 得到的解决方案质量一般。基于深度学习的求解旅行商问题的方法根据构造解决方案方式的不同可以分为学习构造启发式和学习改进启发式。学习构造启发式是以自回归的方式序列式地逐步构造出解决方案, 不需要进一步的改进。Ptr-Net^[7]是第一个采用循环神经网络结构以监督的方式进行求解旅行商问题的神经网络。然而, 这种方法受限于标签数据的质量。为了克服这一限制, 后续研究如 Irwan Bello 的工作^[8]采用了强化学习方式训练模型, 并引入搜索策略来提升求解质量。此外, 基于 Transformer^[9]的注意力模型如 Wouter Kool 提出的 Attention Model^[10], 通过强化学习的 REINFORCE 方法训练, 在小型 TSP 实例上显示出竞争力。文献 [11] 采用了可逆残差网络的方式实现多头自注意力和多指针网络, 以减少内存的消耗, 可以将模型推广到多达 500 个节点的实例上。

学习改进启发式是在一个初始解的基础上进行迭代搜索, 从而达到优化解的目的。文献 [12] 提出了一种基于 2-opt 操作的局部搜索启发式算法, 通过使用深度强化学习算法训练策略, 选择适当的 2-opt 操作符来不断改进当前的解决方案。这种方法通过学习最佳的操作序列, 能够在解决旅行商问题时获得接近最优的解决方案, 然而需要花费大量的时间。文献 [13] 提出了 DeepACO, 一种神经增强 (Neural Enhanced) 的蚁群算法。DeepACO 利用图神经网络生成蚁群算法中的启发式度量, 并结合局部搜索进行解的优化。实验结果表

明, DeepACO 作为一种神经增强的元启发式算法, 在多种路由问题上表现优异。然而, 由于 DeepACO 采用强化学习的方式进行训练, 在训练过程中存在振荡、难以收敛的问题。对此, 文献 [14] 采用了生成流神经网络的方式进行训练, 解决收敛问题的同时, 得到了比 DeepACO 更好的效果。一般来说, 学习改进启发式得到的解决方案要好于学习构造启发式得到的解决方案, 但是解决方案的质量严重依赖选择的搜索方法和迭代次数, 所以需要专业知识和更长的求解时间, 这限制了它们在时间敏感任务中的应用。

大量的研究工作基于循环神经网络、注意力机制, 也有一些工作使用图神经网络^[2]在图上进行求解。但是大多数的工作只考虑节点, 并没有考虑节点之间的边, 直接将欧氏距离作为两点间的路径距离。而现实生活中, 两节点之间的路径距离大多不是欧氏距离, 而且在对三维结构进行视觉检测时, 一些节点之间的路径会与目标结构发生碰撞不能通过, 因此, 会赋予该路径一个比较大的权重^[15]。所以在图上同时考虑节点和边进行旅行商问题的求解更为合理。

图神经网络^[16-17]是专门设计用来处理图数据结构的神经网络, 近年来得到了大量的研究。但是由于图神经网络中的消息传递表达能力在 1 跳聚合下无法区分非同构图。为了解决这一问题, 研究者们借鉴了在自然语言处理和计算机视觉领域取得巨大成功的 Transformer 模型, 并将其拓展到图神经网络中^[18-19], 称为 Graph Transformer。通过允许节点关注图中的所有其他节点, 缓解了有关稀疏消息传递机制的基本限制, 例如, 过度平滑、过度压缩和表达性边界, 在众多图任务中取得了出色的表现。因此, 本文提出了一种基于 Graph Transformer 的无人机全覆盖路径规划求解方法, 称为 GraphFormer。该网络模型为编码器—解码器结构, 编码器结合图卷积和注意力机制提取更加丰富的特征, 解码器预测每一条边在解中的概率。通过搭配搜索和局部优化的方式避免选择权重过大的边, 得到路径更短的解。

1 基于 Graph Transformer 的求解方法

本文将无人机全覆盖路径规划问题视为三维旅行商问题, 同时将本文方法应用在二维旅行商问题上。旅行商问题, 即给定一组节点坐标, 以及两两节点之间的距离, 需要找到一条能够遍历所有节点并回到起始节点的最短路径。形式上表示为一个无向的全连接图 $G < V, E >$ 。 $V = \{v_1, v_2, \dots, v_n\}$ 为节点集合, 表示 n 个节点。 $E = \{e_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq n\}$ 为边的集合, e_{ij} 表示节点 v_i 到节点 v_j 之间的距离, 即节点 v_i 到节点 v_j 的所需的路径成本, 表示为 $cost(v_i, v_j)$ 。路径成本既可以

为节点间的欧氏距离也可以为非欧氏距离。

对无人机进行全覆盖路径规划时, 将无人机的飞行路径限制在图 $G < V, E >$ 中。首先, 将任意两个节点用直线进行相连, 并和待检测三维结构进行碰撞检测。如果节点间连线穿过待检测三维结构或者连线与待检测三维结构之间的距离小于安全距离时, 该路径不可行, 赋予该路径成本一个比较大的权重 W 。为此, 引入邻接矩阵 $\mathbf{m} \in \{0, 1\}^{n \times n}$, 当 $m_{ij} = 1$ 时, 表示节点 v_i 和节点 v_j 之间路径可行, 否则, 表示路径不可行。所以, 路径成本 $cost(v_i, v_j)$ 计算如下:

$$cost(v_i, v_j) = \begin{cases} d_{ij}, & m_{ij} = 1 \\ W, & m_{ij} = 0 \end{cases} \quad (1)$$

其中: d_{ij} 表示节点 v_i 和节点 v_j 之间的欧氏距离, W 为一个比较大的整数。

节点的一个排列也就是旅行商问题实例的一个解称为 tour, 用字母 π 表示。路径的成本 $L(\pi)$ 为:

$$L(\pi | g) = \sum_{i=1}^{N-1} cost(\pi_i, \pi_{i+1}) + cost(\pi_N, \pi_1) \quad (2)$$

所以需要求解的最短路径表示为:

$$\pi^* = \operatorname{argmin}_{\pi \in S} L(\pi | g) \quad (3)$$

S 为旅行商问题实例 g 的可行解集合, 即所有节点的排列组合, 具体与问题实例 g 有关。

本文采用学习改进启发式的方式进行旅行商问题的求解, 提出的 Graph Transformer 模型包括一个编码器和解码器, 以非自回归的方式预测全连接图中每一条边在最优解中的概率, 并采用监督的训练的方式。

1.1 编码器

在图神经网络中, 节点和边的初始嵌入特征通常通过线性映射得到。接下来, 编码器通过图卷积层进行信息传递和特征聚合。图卷积层利用邻接节点的信息来更新节点的表示, 使得每个节点的特征不仅仅依赖于自身, 还能够考虑到图中与其相连的节点。随着层数的增加, 节点的特征会越来越丰富, 逐渐捕捉到图的全局结构和局部模式。

同时结合全局注意力来进一步增强节点表示的能力。全局注意力允许模型根据各节点之间的相关性动态地调整信息传递的权重, 使得模型能够在节点间更加精细地分配注意力, 从而获得更高质量的特征表示。通过图卷积和全局注意力机制的结合, 编码器能够有效地学习到图的深层次语义信息, 网络模型如图 1 所示。

1.1.1 输入层

通过线性层将节点坐标 x_i 和边 ϵ_{ij} 嵌入到 h 维度特征, v_i 为节点 i 的坐标, 其特征表示为 x_i , ϵ_{ij} 为节点 i 和节点 j 之间的边, 即 $cost(v_i, v_j)$ 对应的值, 其特征表示为 e_{ij} :

$$x_i = A_1 * v_i + b_1 \quad (4)$$

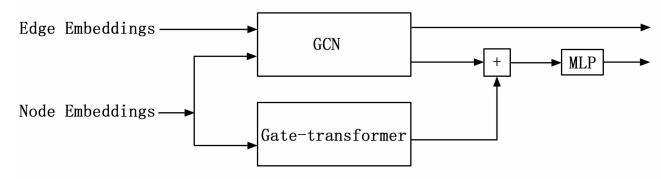


图 1 GraphFormer 网络结构

$$e_{ij} = A_2 * \epsilon_{ij} + b_2 \quad (5)$$

其中: $A_1 \in \mathbb{R}^{h \times n}$ 、 $A_2 \in \mathbb{R}^{h \times 1}$ 、 b_1 、 b_2 为线性层的参数, n 为节点坐标的维度。

1.1.2 图卷积层

\mathbf{x}_i^l 和 \mathbf{e}_{ij}^l 表示节点 x_i 和边 e_{ij} 在第 l 层的特征向量。受文献 [20] 启发, 本文提出一种新的图卷积, 计算如下:

$$\begin{aligned} x_i^{l+1} &= x_i^l + \\ &\text{RELU}\{BN\{x_i^l + \sum_{j \sim N_i} \gamma [\alpha(e_{ij}^l) \odot \beta(x_j^l)]\}\} \quad (6) \\ e_{ij}^{l+1} &= e_{ij}^l + \\ &\text{RELU}[BN(W_1^l * e_{ij}^l + W_2^l * x_i^l + W_3^l * x_j^l)] \quad (7) \end{aligned}$$

其中: N_i 为节点 x_i 的邻居节点, α 、 β 、 γ 为基于多层感知机实现的非线性变换, \odot 为哈达玛积, BN 为批归一化, RELU 为激活函数, W_1 、 W_2 、 W_3 为线性变换。为了减少计算量, 对于有 N 个节点的全连接图, 每个节点只计算 $N/2$ 个邻居节点, 并通过 K 近邻算法在成本矩阵上确定前 $N/2$ 个邻居节点。

1.1.3 Gated Transformer

节点通过图卷积可以聚合局部邻居节点的特征, 同时结合全局自注意力, 使节点具有全局的视野, 使得每个节点能够在聚合邻居信息时, 考虑到全图其他节点的重要性, 提升了模型处理全局结构信息的能力。同文献 [21], 首先对节点的特征进行批量归一化操作, 通过多头注意力 (MHA, multi-head attention) 进行自注意力的计算, 计算如下:

$$Y = MHA[BN(X)] \quad (8)$$

然后, 并没有直接采用残差连接, 而是通过门控循环单元 (GRU, gate recurrent unit) 进行最终的特征更新, 计算如下:

$$r = \sigma(W_r^l * y + V_r^l * x) \quad (9)$$

$$z = \sigma(W_z^l * y + V_z^l * x - b_z^l) \quad (10)$$

$$h = \tanh[W_g^l * y + V_g^l * (r \odot x)] \quad (11)$$

$$g^l(x, y) = (1 - z) \odot x + z \odot h \quad (12)$$

1.1.4 Graph Transformer 层

X^l 通过图卷积层和多头自注意力层得到 X_M^{l+1} 和 X_T^{l+1} , 将两者进行相加, 然后通过前馈神经网络得到更新后的特征表示, 计算如下:

$$X^{l+1} = W_f^l (X_M^{l+1} + X_T^{l+1}) \quad (13)$$

为了减少模型的复杂度, 边特征的更新并没有通过多头注意力层, 考虑到节点在更新特征后, 已经整合了全图的特征。所以, 在更新边的特征时直接使用更新过的节点的特征, 边特征的更新计算如下:

$$e_{ij}^{l+1} = e_{ij}^l + \text{RELU}[BN(W_1^l * e_{ij}^l + W_2^l * x_i^{l+1} + W_3^l * x_j^{l+1})] \quad (14)$$

以上, 组成了 Graph Transformer 模型。

1.2 解码器

本文采用非自回归的方式直接预测每一条边在最优解中的概率, 解码器由多层感知机组成, 对边进行二分类, 采用加权二元交叉熵作为损失函数。由于随着问题规模的增大, 旅行商问题解对应的路径中的负类(即边不在解中的情况)数量会远大于正类(即边在解中的情况)^[2]。这种类别不平衡会导致模型偏向预测负类。因此, 为了平衡这种类别不平衡, 需要对正负类进行适当的加权, 使得模型能够更好地学习到正类的特征。正类、负类的权重计算分别如下:

$$w_1 = \frac{n^2}{2n * c} = \frac{n}{2c} \quad (15)$$

$$w_0 = \frac{n^2}{(n^2 - 2 * n) * c} = \frac{n}{(n - 2) * c} \quad (16)$$

对于每一条边 e_{ij} , 通过 Softmax 函数将模型的输出 e_{ij0} 、 e_{ij1} 转化为两个类别对应的概率 \hat{p}_{ij0} 和 \hat{p}_{ij1} , 计算如下:

$$\hat{p}_{ij0} = \frac{e^{z_{\phi_0}}}{e^{z_{\phi_0}} + e^{z_{\phi_1}}}, \quad \hat{p}_{ij1} = \frac{e^{z_{\phi_1}}}{e^{z_{\phi_0}} + e^{z_{\phi_1}}} \quad (17)$$

其中: \hat{p}_{ij1} 、 \hat{p}_{ij0} 分别表示模型预测的节点 v_i 和节点 v_j 之间存在边和不存在边的概率。然后, 使用加权交叉熵来计算边 e_{ij} 的损失:

$$L_{ij} = -w_{y_{ij}} * \log(\hat{p}_{ij, y_{ij}}) \quad (18)$$

其中: y_{ij} 为边 e_{ij} 对应的真实标签, $w_{y_{ij}}$ 、 $\hat{p}_{ij, y_{ij}}$ 分别为类别 y_{ij} 对应的权重和预测概率。对于有 N 个节点的全连接图, 有 N^2 条边, 整个批次的损失是所有边的加权交叉熵损失的平均值, 计算如下:

$$L = \frac{1}{B * N^2} \sum_{b=1}^B \sum_{i,j} -w_{y_{ij}} * \log(\hat{p}_{ij, y_{ij}}) \quad (19)$$

其中: B 为批次的大小。

1.3 波束搜索和局部优化

经过解码器得到实例的概率热力图, 一个可行解 π 的概率为:

$$p(\pi) = \prod_{i=1}^{N-1} \hat{p}_{i, i+1, 1} \quad (20)$$

其中: $\hat{p}_{i, i+1, 1}$ 表示节点 i 和其下一个节点之间存在边的概率。如果将一个节点作为起始节点, 并根据边存在的概率每次贪婪地选择概率最大的下一个节点, 这很可能导致无效的行程。因此, 本文采用了波束搜索^[22]

方法, 从概率热力图中构造有效的行程, 同时考虑到非欧氏距离的情况, 结合 2-opt 局部优化算子对解进行进一步优化, 从中选取路径最短的作为最终解。

2 数据集和实验参数设置

2.1 数据集及评价指标

在实验中, 从单位平方 $[0,1]^2$ 中均匀采样生成随机实例, 随机生成 100 000 个实例作为训练集。对比实验中, 测试数据包括随机生成的实例和 TSPLIB 库中的实际实例。TSPLIB 是一个著名的 TSP 库, 它包含 100 个具有不同节点分布的实例。这些实例来自现实生活, 节点个数的范围为 14~85 900, 本文从中选取一些实例进行实验, 节点数量最多为 500 个。在二维场景的实验中, 精确求解器 Concorde 得到的解 L 作为实例的最优解, 使用模型得到的解为近似解 \hat{L} 。评价指标为路径长度和相对误差 Gap 。长度单位为 m, 用 Len 表示, Gap 定义如下:

$$Gap = \frac{\hat{L} - L}{L} \quad (21)$$

2.2 实验参数设置

使用一个 RTX 2080 Ti x2 GPU 来训练模型, 在每个 epoch 中, 随机选取 10 000 个实例进行训练, 批量大小为 20, 使用 Adam 作为优化器, 初始学习率为 10^{-3} , 学习率衰减同文献 [2]。编码器一共包含 6 层, 初始特征嵌入维度为 128, 所以多头注意力中 $d_k = 128, h = 8$ 。通过实验验证, 较宽的波束搜索能够提高解的质量。然而, 较大的波束宽度会增加计算时间, 因此为了平衡求解质量和推理速度, 波束搜索的宽度设置为 1 280。

3 二维路径规划实验结果及分析

为了验证提出方法的性能进行了大量的实验, 并与一些深度学习求解方法进行了对比。二维路径规划中, 实验涵盖了 3 种基准任务(20、50、100 节点的旅行商问题), 包含欧氏距离和非欧氏距离的情况, 同时使用 TSPLIB 库中的实例验证网络模型的泛化能力。

二维路径规划中, 进行对比实验的方法包括端到端的基于强化学习的算法: AM^[10]、POMO^[3]、Point-erformer^[11], 以及基于搜索的算法: GCN^[2]、DRL + 2-opt^[12]。

在欧氏距离情况下, 对这 3 种基准任务进行实验的测试数据同文献 [11~12], 每个任务都包含了 10 000 个测试实例。实验结果如表 1 所示, 从表中可以看出本文模型在 TSP_20 这个基准任务上得到的解要好于精确求解器得到的解。同时在 TSP_50 任务上, 相比于其他网络模型求得的路径长度最短。但随着问题规模的增大, 搜索空间变大, 得到更好的解的难度增大, 但依然能够得到具有竞争力的解。

表 1 随机生成实例对比

方法	TSP_20		TSP_50		TSP_100	
	Len/m	Gap/%	Len/m	Gap/%	Len/m	Gap/%
最优解	3.830 6	—	5.691 8	—	7.764 5	—
AM	3.841 3	0.279	5.784 9	1.636	8.100 8	4.331
GCN	3.835 1	0.117	5.745 1	0.936	10.736 5	38.277
POMO	3.831 3	0.018	5.697 2	0.095	7.789 7	0.325
Pointerfomer	3.832 5	0.050	5.693 6	0.032	7.773 3	0.113
DRL+2-opt	3.831 8	0.031	5.698 2	0.112	7.820 6	0.723
GraphFormer	3.830 3	—0.008	5.691 9	0.002	7.780 8	0.210

为了进一步验证本文模型的泛化性, 选取 TSPLIB 中的实例进行了对比实验, 并且考虑了每个网络模型的运行时间, 使用 *Time* 表示, 单位为 s。将每个测试实例的执行时间进行相加之后得到的均值作为最后的执行时间。所用的测试实例节点数量范围为 1~500。实验结果如表 2 所示, 在 TSP_101~200 和 TSP_201~500 上的实验结果均是最佳的, 所以 Graph Transformer 模型有很强的泛化性, 但是随着实例节点数量增加, 也需要更长的求解时间。

表 2 TSPLIB 实例对比

方法	TSP_1~100		
	Len/m	Gap/%	Time/s
最优解	19 454.167	—	—
AM	21 235.002	9.154	2.57
GCN	28 296.533	45.452	8.40
POMO	19 638.867	0.949	1.34
Pointerfomer	19 726.159	1.398	4.02
GraphFormer	19 777.166	1.66	4.24
方法	TSP_101~200		
	Len/m	Gap/%	Time/s
最优解	29 605.222	—	—
AM	34 345.577	16.012	2.61
GCN	42 123.876	42.285	5.98
POMO	30 759.594	3.899	1.38
Pointerfomer	30 401.263	2.689	5.33
GraphFormer	30 361.876	2.556	7.33
方法	TSP_201~500		
	Len/m	Gap/%	Time/s
最优解	48 084.375	—	—
AM	61 742.820	28.405	2.81
GCN	72 847.353	51.499	7.47
POMO	55 110.572	14.612	1.53
Pointerfomer	52 138.421	8.431	5.53
GraphFormer	49 647.684	3.251	17.07

同时考虑到旅行商问题的一些特殊情况, 即节点之间的边不为欧氏距离的情况。由于只考虑节点坐标的对比算法不能处理这种情况, 所以本文比较了图神经网络算法、最大最小蚁群算法^[23]和精英遗传算法^[24]。为了保证结果的公平性, 在模型训练的时候并

没有添加对抗样本。每个基准任务包括 100 个随机生成的测试实例, 每个实例的距离矩阵随机生成。最大最小蚁群算法参数设置为蚂蚁数量为 150, 信息素重要性因子 $\alpha = 1$, 启发信息重要性因子 $\beta = 2$, 信息素挥发率为 0.3, 信息素增加强度系数为 5, 迭代次数为 500。精英遗传算法参数设置为种群数量为 150, 变异率为 0.05, 精英率为 0.4, 迭代次数为 10 000。实验结果如表 3 所示, 在时间方面, 使用神经网络的方法远远优于最大最小蚁群算法和精英遗传算法。相比于 GCN 网络模型, 最大最小蚁群算法和精英遗传算法能够得到长度更短的解, 而本文算法得到的解长度最短。所以从实验结果可以看出本文算法无论是在路径长度和求解时间方面均具有优越性。

表 3 非欧氏距离对比

方法	TSP_20		TSP_50		TSP_100	
	Len/m	Time/s	Len/m	Time/s	Len/m	Time/s
最大最小蚁群算法	4.135 3	14.093	6.675	82.642	9.871 1	324.665
精英遗传算法	4.220 5	67.655	7.548 6	125.606	12.909 8	259.672
GCN	4.538 2	0.131	8.429 6	0.237	11.256 9	0.463
GraphFormer	4.078 7	0.592	6.394 1	3.268	9.434 0	10.123

为了验证编码器中图卷积和自注意力结合的有效性, 在包含 100 个节点的实例上进行了消融实验, 包括欧氏距离、非欧氏距离两种情况。消融实验对比了编码器包含自注意力和不包含 (表示为 w. o. MHA) 两种情况, 对比实验结果见表 4。从表中可以看出图卷积和自注意力结合在两种情况的路径规划中均能得到路径长度更短的解, 所以二者结合可以起到积极的作用。

表 4 消融实验结果

方法	欧氏距离	非欧氏距离
	Len/m	Len/m
w. o. MHA	8.046 7	10.538 1
GraphFormer	7.806 8	9.503 2

4 无人机全覆盖三维路径规划实验及分析

三维路径规划中, GraphFormer 模型中节点的坐标为三维, 其余设置与二维路径规划中相同。在节点数为 50 的旅行商问题上进行对比实验。针对三维路径规划的情况, 由于其它的对比模型仅仅适用于二维情形, 所以对比实验中只对比了本文算法、最大最小蚁群算法和精英遗传算法, 同时考虑了欧氏距离和非欧氏距离两种情况。

在随机生成的 10 个实例上进行了对比实验。对每个测试实例, 最大最小蚁群算法和精英遗传算法各执行 10 次, 取平均值作为结果。表 5 和表 6 分别对应欧氏距离和非欧氏距离情况下的实验结果。可以看出本文算

法在两种情况下不仅能得到最好的结果, 而且求解时间远远低于最大最小蚁群算法和精英遗传算法。

表 5 三维欧氏距离对比

方法	TSP_50	
	Len/m	Time/s
最大最小蚁群算法	11.558	85.794
精英遗传算法	11.438	126.083
GraphFormer	11.128	7.523

表 6 三维非欧氏距离对比

方法	TSP_50	
	Len/m	Time/s
最大最小蚁群算法	6.463	84.796
精英遗传算法	7.355	125.851
GraphFormer	6.238	4.297

三维空间路径规划的一个重要应用为使用无人机对三维结构进行损伤检查, 如大型客机等。在路径规划过程中, 要避免无人机与待检测三维结构之间的碰撞。在将本文算法应用于飞机表面全覆盖检测时, 首先进行碰撞检测, 得到代价矩阵 *cost*, 然后将其和节点一起输入 GraphFormer 模型中, 得到路径规划的结果。

在全覆盖路径规划实验中, 将待检测的飞机模型归一化到单位球中, 通过解决视图规划问题, 在满足覆盖要求的情况下, 需要遍历 60 个节点。

最大最小蚁群算法和精英遗传算法各运行 10 次取最佳的结果。实验结果如表 7 所示, 最大最小蚁群算法的平均结果为 11.67 m, 精英遗传算法的平均结果为 19.79 m, 并且最大最小蚁群算法求得的最优解优于精英遗传算法, 而本文算法能得到最优的解, 并且用时远远低于二者。所以本文算法在路径长度和计算时间上均占有优势, 在三维路径规划中性能优异。

表 7 全覆盖路径规划结果

方法	Len/m	Time/s
本文算法	11.14	5.397
最大最小蚁群算法	11.57	121.0
精英遗传算法	18.11	166.670

图 2 展示了表 7 中本文算法、最大最小蚁群算法和精英遗传算法规划的最佳路径。从图中可以观察到, 本文算法规划的路径相对简洁, 最大最小蚁群算法规划的路径在飞机头部区域, 本文算法规划的路径相同, 而精英遗传算法则得到了一条较为杂乱的路径。

5 结束语

本文针对无人机全覆盖路径规划问题, 提出了一个基于 GraphFormer 的模型, 并且适用于二维路径规划中。实验证明该方法无论是在二维还是在三维路径规划中都有着不错的表现。在 TSP_20 和 TSP_50 任务中,

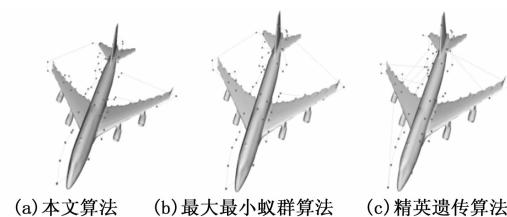


图 2 全覆盖路径规划结果

相较于其它基于深度学习的模型, 表现出最佳的性能, 并且在 TSP_20 任务上能获得比精确求解器更优的解。在 TSPLIB 上的实验结果表明, 该模型具有良好的泛化能力, 并且能够适应非欧氏距离。本文算法在无人机的全覆盖路径规划中性能表现优异, 然而, 本文算法在实例节点数量增加时, 求解时间也会随之增长。因此, 未来的工作将专注于优化模型的搜索策略, 旨在提高解的质量的同时, 缩短求解时间, 以应对更大规模问题的挑战。

参考文献:

- [1] 陈丽, 陈洋, 杨艳华. 面向三维结构视觉检测的无人机覆盖路径规划 [J]. 电子测量与仪器学报, 2023, 37 (2): 1–10.
- [2] JOSHI C K, LAURENT T, BRESSON X. An efficient graph convolutional network technique for the travelling salesman problem [J]. Arxiv, 2019-06-04.
- [3] 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.
- [4] HELSGAUN K. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems [D]. Roskilde: Roskilde University, 2017.
- [5] ALKAYA A F, DUMAN E. Application of sequence-dependent traveling salesman problem in printed circuit board assembly [J]. IEEE Transactions on Components, Packaging and Manufacturing Technology, 2013, 3 (6): 1063–1076.
- [6] ANGENIOL B, VAUBOIS G D L C, LE TEXIER J Y. Self-organizing feature maps and the travelling salesman problem [J]. Neural Networks, 1988, 1 (4): 289–293.
- [7] VINYALS O, FORTUNATO M, JAITLY N. Pointer networks [C] //Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2. 2015: 2692–2700.
- [8] BELLO I, PHAM H, LE Q V, et al. Neural combinatorial optimization with reinforcement learning [J]. Arxiv Preprint Arxiv: 1611.09940, 2016.

(下转第 277 页)