# 基于GNN的无人机网络优化

## 摘要

## 0 引言

随着无线通信技术的不断进步，无人机（UAV）在无线通信和计算领域展现出了巨大的潜力。无人机的高机动性允许它们快速响应各种任务需求，特别是在灾难恢复、临时事件覆盖或偏远地区通信服务中显示出其独特优势。无人机集群不仅能够作为移动边缘服务器在不同地点提供计算资源，还能作为中继节点扩展和增强地面基站的通信范围。此外，与传统的基站建设相比，无人机集群的成本低，机动性高，为实现快速部署和灵活配置提供了可能。

在无人机在无线通信领域广泛应用的同时，如何优化部署方案、任务分配、资源配置等方面的问题也为研究人员提出了新的挑战。一些工作提出了多种解决方案。例如，基于进化算法（Evolution Algorithm，EA）的方案通过模拟自然进化机制来优化网络配置，基于深度神经网络（Deep Neural Networks，DNN）的方案能够通过学习复杂的网络特征来预测最优配置。但不同于地面基站，由于无人机的高机动性，网络的通常具有高动态性和复杂服务需求，特别是在网络规模和拓扑结构不断变化的情况下，进一步提出了可扩展性的需求。然而目前的研究方案，存在大规模网络中收敛速度慢，难以实时响应网络动态变化，训练成本高，面对网络结构变化时泛化能力有限等问题。无人机网络规模的扩大和拓扑结构的动态变化主要带来了两个方面的动态性：观测空间的动态性和决策空间的动态性。观测空间的动态性主要来自于环境因素的动态变化和部分观测范围的限制。比如对于无人机作为边缘服务器的场景，无人机的移动会改变边缘计算资源的分布，每个用户的可观测利用的边缘服务器也会相应改变。当决策取决于无人机与其余环境实体之间的关系时，动态的链接会导致无人机节点的决策空间动态性，比如无人机的随机加入和退出，以及用户的移动，会导致原有的通信链路断开和新的链路形成。一些传统的资源分配和网络管理策略面对观测空间和决策空间的动态性难以适应。如基于进化算法和凸优化的方案，当网络中无人机或者用户位置发生即使微小的变化，也需要重新迭代出最优方案，存在计算复杂度过高的问题。基于深度神经网络的方案可以通过预训练的方式避免重复迭代问题，因此本文主要探索基于深度学习的方案在无人机网络中的应用，但是一般的神经网络结构如多层感知机（Multilayer Perceptron， MLP）受限于基于向量化的信息编码方式，难以适应观测空间和决策空间的维度变化。

近年来，基于图神经网络（Graph Neural Networks，GNN）的方案在无线通信领域应用广泛。GNN以其对图结构数据的强大处理能力，从网络结构上实现了能够适应动态的图节点数量，并且可以有效提取图的拓扑信息。基于此，本文针对无人机网络，提出了基于GNN的方案架构以适应无人机网络中观测空间的动态性以及决策空间的动态性。针对观测空间的动态性，首先对无人机网络进行了有向图建模，将节点分为无人机节点与环境实体节点，其中环境实体节点包含环境中可以提供信息的非无人机节点。其次我们设计了基于GNN的编码方式以及机器学习算法框架。对于决策空间的动态性，本文提出了基于边决策的GNN模型，通过图建模和边权重的拟合，使得GNN能够适应动态的决策空间。最后，本文提出了两个无人机网络的案例研究，证明了本文所提出的基于GNN的方案的有效性。

## 无人机网络动态观测模型

在无人机网络中通常需要通过观察其他无人机和其他实体因素，如障碍物，用户等环境因素进行决策。然而决策过程中对于每个无人机如何有效提取所观测的环境因素特征是一个重要挑战，一些研究采用固定向量进行编码作为环境特征，但是这种编码方式无法适应环境实体因素的增加和减少，固定长度向量编码方式导致的信息缺失或信息杂糅不可避免。本文提出基于GNN的动态观测特征模型，首先构建动态图模型，然后采用GNN框架为每个无人机编码和提取环境信息特征，最后基于机器学习的框架进行决策。

### 1.1 基于GNN的动态观测网络编码

针对无人机网络，我们首先定义动态观测图模型为图，对应一个时间步，定义：

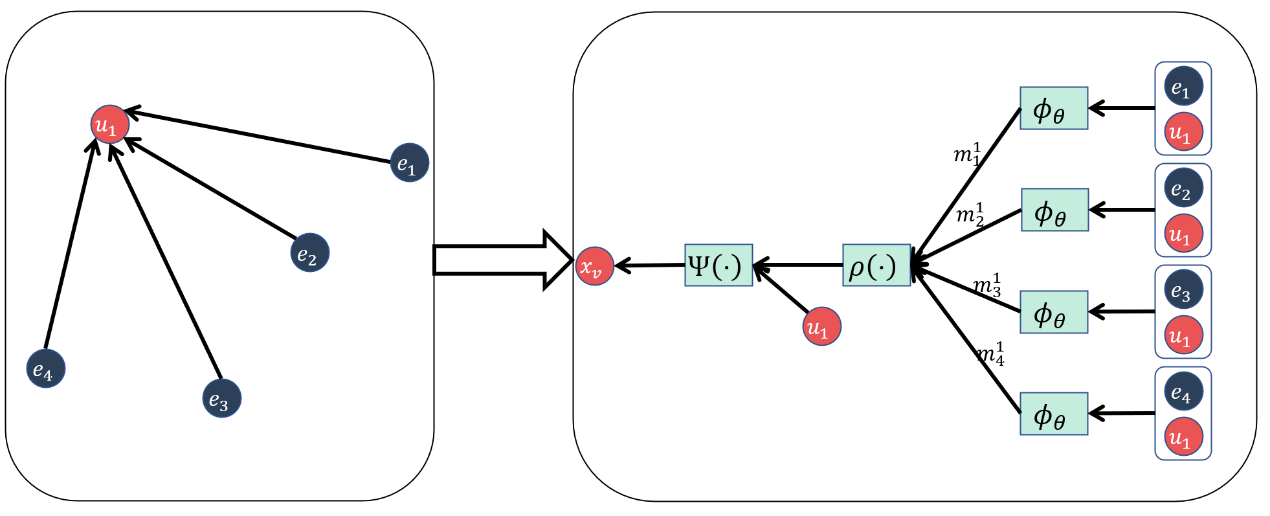
* 表示节点集，包括无人机节点和可观测的实体节点，每个节点具有特征。
* 是边集，包括无人机节点间的边和无人机节点与实体节点之间的边。
* 是无人机节点间的邻接矩阵，其中表示无人机与无人机在时间具有直接的通信链路。
*  是无人机与实体节点间的邻接矩阵，其中表示无人机在时间可以观测到实体节点。

图 1 图神经网络架构

当无人机节点集合和实体节点集合的拓扑连接关系变化时，邻接矩阵、会相应变化，当、变化时，邻接矩阵、的维度也会相应改变。为了适应动态的环境实体因素，本文提出采用GNN进行编码的方案。GNN的主要目的是通过从相邻节点提取信息来研究节点特征，通过边传播和聚合特征来学习节点的表示。其中GNN结构中最重要的部分是信息传递机制，包括消息传播和消息聚合，如图 1所示，表示无人机节点，表示环境中其他实体节点，无人机和实体节点之间通过有向边连接，具体消息传递与聚合方式如式与式：





其中表示节点的邻居节点集合。是第层中节点的特征，而表示一个邻居节点的特征。表示参数化的消息传播函数，确定了从邻居传递到节点的消息传递，对于所有可以进行消息传递的两个节点、之间，函数具有相同的参数。表示从邻居节点传播到的消息。在式（）中，表示聚合函数，用于聚合所有邻居节点的消息，聚合函数通常设置为，，等，假设为维向量，集合按照第一维采用聚合函数，得到聚合特征也为维向量。特征更新函数根据特征和聚合的消息更新节点的特征，同消息传播函数，对于所有节点特征更新函数具有相同的参数。

### 1.2 基于GNN编码的机器学习框架

在现代机器学习框架中，预训练已成为一种普遍的做法，尤其在处理复杂任务或数据稀缺情况下。预训练涉及在一个大型且多样化的数据集上训练模型，以学习通用的特征表示。这些表示随后可以在具体任务的数据上进行微调，以实现快速适应和性能提升。以下是如何将可预训练的概念融入到有标签学习、无监督学习和强化学习中。

在有标签学习中，预训练通常涉及在一个大规模的、标注好的数据集上进行模型训练。例如，在无人机网络中，可以使用各种地形和条件下的历史飞行数据来训练GNN模型。这样预训练的模型可以捕捉到环境特征与决策标签之间的复杂关系。当模型部署到新的无人机网络任务时，预训练的模型可以通过少量的微调迅速适应新环境。这种方法可以表示为：



其中， 是预训练后的模型参数， 是预训练数据集的大小， 和  分别是预训练数据集中的标签和观测图。

无监督学习可以利用大量未标记的数据来进行预训练。例如，可以使用无人机在不同环境中采集的大量观测图，通过自编码器学习这些图的潜在表示。这些表示可以捕获环境的关键特征，而不依赖于外部的标签信息。在具体任务上，这些预训练的表示可以用于初始化模型，加快后续学习过程，并提高模型的泛化能力。数学上，这可以表示为：



其中， 和  分别是预训练后编码器和解码器的参数。

强化学习可以通过在模拟环境中预训练智能体来学习策略。这种预训练策略可以使智能体在实际部署前就掌握一定的基础策略，降低实际环境中探索的风险和成本。预训练的强化学习模型可以在新环境中快速调整其策略，以适应新的任务要求。这可以通过以下优化问题来定义：



其中， 是预训练后的策略参数， 是预训练阶段的时间地平线， 和  分别是预训练阶段的状态和动作。

预训练不仅提高了模型的泛化能力，还显著减少了在线学习或微调时的计算负担。这种方法特别适用于动态变化的无人机网络环境，其中模型需要快速适应新的任务和条件。通过预训练，无人机网络的决策系统可以在新场景中直接通过推理以较低的复杂度获取最优策略。

同时，对于一个包含V个节点和E条边的图，一个简单的GNN层的计算复杂度是。实际上，计算复杂度还取决于节点特征的维度和模型的深度。因此，一个GNN的计算复杂度可以近似为。预训练可以减少在线学习阶段所需的计算资源，因为微调预训练模型通常比从头开始训练一个全新的模型需要较少的训练迭代。这对于资源受限的无人机网络环境尤为重要。

## 无人机网络动态决策空间模型

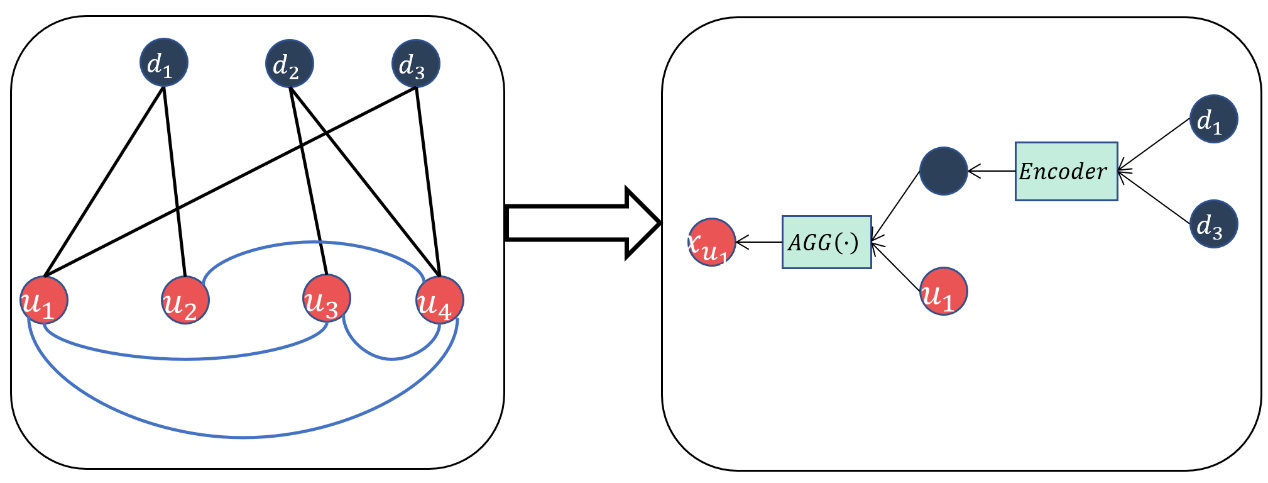
基于GNN的决策框架直接根据节点自身及其邻居节点的特征信息输出决策信息，决策空间的维度与网络规模和网络拓扑无关。然而存在一些特定的任务场景，其决策空间与网络拓扑有关，或者每个节点的决策空间维度与拓扑结构有关，无法直接通过节点聚合的信息进行决策。这一部分通过构建无人机网络的图模型，并引入基于边决策的GNN框架，以适应决策空间的动态变化。

图 2 边决策图神经网络架构

### 2.1 无人机网络决策图建模

如图 2所示，将图 2中的节点分为智能体节点集合和决策节点集合，智能体节点集合包含所有能够执行任务的无人机节点，而决策节点集合包含那些用于决策制定的节点，这些节点为智能体提供了执行决策所需的信息和指令。

在这个模型中，智能体节点的决策空间是动态的，依赖于与它相连的决策节点集合的状态，可以表示为，其中，是一个映射函数，表示智能体节点所连接的决策节点集合。决策节点的状态不仅包括其自身的属性，如位置、能量状态等，还包括它与其他节点的连接关系。决策节点集合的状态可以表示为一个向量，对于决策节点，其状态向量，其中和分别代表节点的位置坐标，表示节点的能量状态等信息。

智能体节点的决策空间的维度可以随着决策节点集合的变化而变化，这意味着每当决策节点的状态或拓扑连接发生变化时，智能体节点必须重新评估其可用的决策选项。例如，如果新的决策节点加入或现有的决策节点离开网络，或者决策节点之间的连接关系变化，智能体节点的决策空间  就会相应地调整。

智能体节点和决策节点之间的连接可以用邻接矩阵来表示，其中如果智能体节点与决策节点之间有连接，否则。因此，智能体节点的决策空间可以进一步表示为：



其中，表示连接操作，它将智能体节点所连接的所有决策节点的状态向量组合起来，形成决策空间。

智能体节点的决策空间的动态性是由决策节点集合的状态和拓扑结构的变化所驱动的。无人机网络中的每个智能体节点都必须实时监控其连接的决策节点，并根据这些节点的状态来调整自己的行为和策略。这种动态性要求无人机网络具有快速响应和适应环境变化的能力，以确保任务的顺利完成。

### 2.2 基于边决策的GNN框架

智能体节点集合到决策节点集合的链接变化时，智能体节点的决策空间以及决策空间维度维度也会改变，直接通过node-output GNN可以适应观测空间的变化，但是难以适应决策空间的变化。因此本文提出基于边决策的GNN架构，如图 2所示，根据图的拓扑连接设立智能体节点的决策空间，具体过程如式（）-（）：











其中和分别是智能体节点集合和决策节点集合，是节点的邻接节点集合，当大于0时等于，否则是，其中是一个大于0但小于1的因子。是从邻居节点传递给节点的提取信息，是一个可训练的聚合网络，用于提取邻居节点的特征。 是一个消息连接函数，用于将所有邻接节点的消息压缩成一个向量，通常设置为  或，是评估邻接节点对当前节点影响力的注意力值，而是一个可训练的神经网络，用于从初始输入特征更新节点特征到输出特征。、、和是可训练参数的矩阵。和用于计算注意力值 ， 用于确定智能体节点  到决策节点  的链路权重  和 ， 用于确定服务器  到用户  的链路权重 。以链路权重作为智能体节点的决策输出。

## 案例研究

在本研究中，我们探讨了两种无人机网络优化场景，并采用了动态观测和动态决策的GNN框架，通过仿真实验，我们验证了所提出方案的有效性。首先，考虑到设备对设备（D2D）通信的情境，我们将无人机作为通信中继，并针对无人机的最优部署位置以及在多无人机环境中D2D通信对的路由策略进行了联合优化。为此，我们采用了一个基于GNN的两阶段优化方案，该方案展现了优异的性能。在第二个场景中，无人机被视为边缘服务器，以辅助任务调度和资源分配的优化。在这种配置下，用户将计算任务以不同的比例卸载到无人机边缘服务器，以便进行边缘计算。该过程涉及优化任务分配、计算资源分配和传输功率分配，旨在最小化处理延迟。本研究采用了一个基于边决策的GNN框架对此问题进行建模，并通过无监督学习方法进行训练，最终实现了令人满意的性能表现。

### 2.1 基于GNN的无人机辅助D2D通信的部署与路由联合优化

**2.1.1 系统模型**

这一部分考虑一个无人机集群作为移动中继辅助物联网设备间的数据传输场景。考虑对物联网发送端和接收端，它们通过个无人机进行中继通信，每个无人机的通信范围限定在米内。目标是要求不增加无人机数量的情况下优化无人机的位置部署和D2D通信的路由选择。

假设通信基于正交频分复用(OFDM)，所有设备的发射功率为。由于小尺度信道衰落变化快，本文仅考虑大尺度路径损耗。发送器的流速由以下公式决定：



其中是跳的速率，是节点间距离，是单位距离的路径损耗，是路径损耗指数，是噪声功率。

优化问题是最大化总系统流速：



使用加权无向图表示网络结构，包括物联网设备和无人机，代表无线链路。决策变量包括无人机位置和中继路径集。目标函数和约束条件如下：

目标函数：













其中流量不超过链路到的速率，对于无人机，进入的流量等于离开的流量，确保流量守恒。每个节点作为发送器的中继路径数量限制为1，避免多路径发送。中继路径的选择由流量决定，若流量非零，则路径被选中。速率取决于节点间距离，若距离小于等于，则按照信号衰减计算速率，否则速率为0。

**2.1.2 基于GNN的两阶段方案**

针对案例问题，本研究提出了一种基于GNN的两阶段方案，以确定所有无人机的位置并为每对发送者和接收者选择中继路径。首先，我们设计了一个算法在任意无人机位置下为所有用户选择最优中继路径，然后设计了一个算法来确定无人机位置。选择最优中继路径的问题可以形式化为问题 \ref{P2}。

我们采用两个GNN来解决该问题，一个用于优化所有无人机的位置 ，另一个用于选择中继路径 。通过梯度传播，神经网络能够学习  和  之间的关系并共同优化。训练过程中，我们首先训练一个基于强化学习的中继GNN（RGNN），然后训练一个位置GNN（LGNN）。推理过程中，我们首先使用LGNN优化无人机位置，然后使用RGNN为每对用户选择最佳中继路径。

所提出的两阶段训练和推理框架有四个主要优势：分解问题、加速收敛、提取有价值特征以及提高鲁棒性。根据精确度和计算延迟需求，我们可以用任何架构的神经网络，甚至启发式算法和传统优化算法来替换任何GNN。

* 基于RGNN的中继选择方法

我们提出了一种基于强化学习的GNN方法，即RGNN，以低复杂度获得所有用户的最佳中继路径。RGNN的强化学习框架包括状态、动作和奖励。状态定义为图 ，动作为选择下一个中继节点，奖励为最小化中继路径中最长距离的负值。RGNN的目标函数旨在最小化中继链路中最大距离。RGNN结构包括编码器和解码器，编码器用于提取所有节点和选定中继路径的特征，解码器用于计算每个节点的注意力值并选择下一个中继节点。

* 基于LGNN的无人机位置优化

我们提出了一种基于无监督学习的LGNN训练方法，无需标记数据并具有高鲁棒性。LGNN的结构旨在最大化所有用户对的通信速率，通过LSTM处理两个有通信需求的用户的特征，并使用GNN提取用户和无人机之间的关系。LGNN进一步使用注意力方法计算不同用户对无人机的影响，并通过双向LSTM（BiLSTM）处理UAVs特征，以优化整体性能。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **算法 1**  RGNN与LGNN方法的训练过程 | | | | |
| 1. 参数初始化 2. For epoch do | | | | |
|  | 随机生成一个图以训练RGNN | | | |
|  | For t do | | | |
|  |  | 设，其中，是中去除了不在中的节点所形成的边的子集； | | |
|  |  | 使用RGNN计算中每个节点的嵌入向量； | | |
|  |  | 初始化状态； | | |
|  |  | for t do | | |
|  |  |  | | If |
|  |  |  | | 终止循环 |
|  |  |  | | Endif |
|  |  |  | | 计算中继路径的特征 |
|  |  |  | | 计算  中所有节点的概率 |
|  |  |  | | 选择概率最大的节点作为下一跳中继节点 |
|  |  |  | | 更新， |
|  |  | Endfor | | |
|  |  | 随机梯度下降更新RGNN的参数 | | |
|  | Endfor | | | |
|  | For t do | | | |
|  |  | | 随机生成一个图以训练LGNN | |
|  |  | | 计算所有无人机的位置 | |
|  |  | | 更新LGNN的参数 | |
|  | Endfor | | | |
| 1. Return | | | | |

**2.1.3 仿真结果**

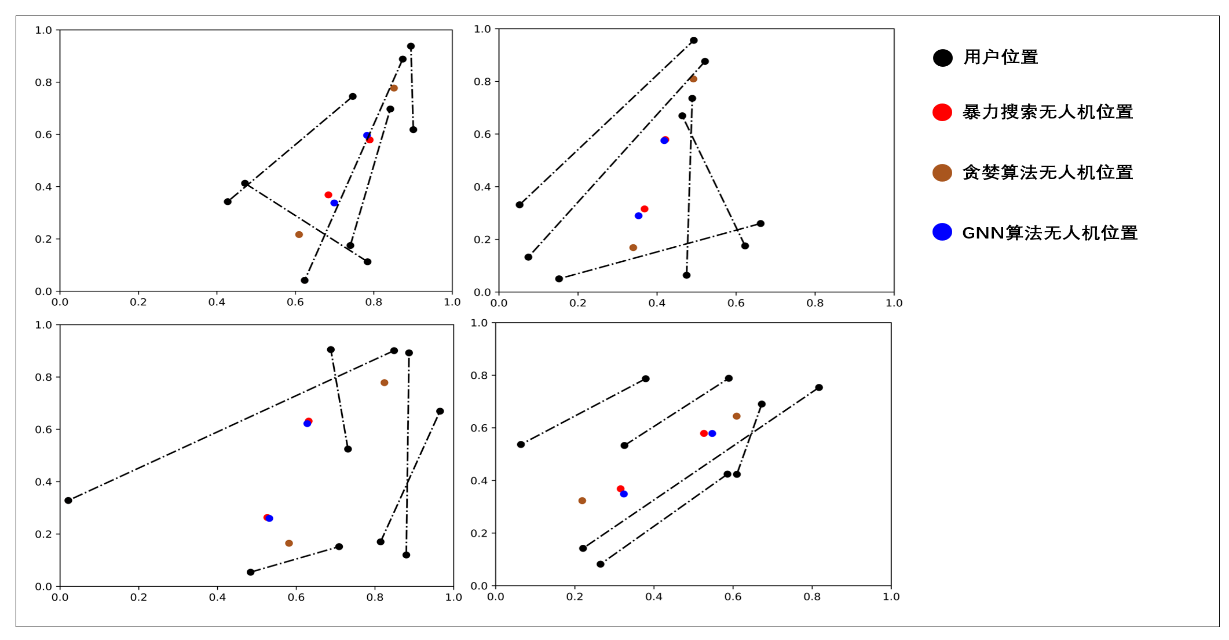
 为了验证本文提出的基于GNN方案得有效性，本部分将本文所提出的方案与暴力搜索方法和基于聚类的贪婪算法进行了比较。在暴力搜索中，整个区域场景被离散化为一个20\*20的网格，每个无人机可以放置在任何网格中。遍历所有可能的无人机位置组合，并采用贝尔曼-福德（Bellman Ford）算法用于计算当前部署位置下的最佳中继路径，并计算每种部署组合的性能表现。选取达到最佳网络性能的无人机位置组合，作为最优解。同时，我们也使用了一种贪婪算法，该算法将所有用户聚类，并将无人机放置在若干个聚类的中心。如图 3所示，水平和垂直坐标值分别代表平面中用户和无人机的坐标位置，我们在其中放置了5对共10个用户设备，被虚线连接的两个用户设备之间存在通信需求，并放置了两个无人机用于辅助中继。在这四个子图中，基于暴力搜索的无人机位置方案可以视为最优方案，从结果可以看出，由贪婪算法优化的位置总是远离由暴力搜索和提出的方法优化的位置，将无人机部署在用户群集的中心并不是最优解。而所提出的基于GNN的方案非常接近于暴力搜索解， 说明本文所提出的方法可以以较低的复杂度得到近似最优解。

图 3 仿真结果

### 基于边决策的无人机边缘服务器辅助的任务调度与资源分配优化方案

### 2.2.1 系统模型

这一部分我们研究了一个多用户设备与多无人机边缘服务器的网络。在一定区域内，个用户设备随机分布，每个用户有大小为的数据任务需处理。由于设备计算资源限制，任务需卸载至无人机边缘服务器处理。网络中还有个无人机边缘服务器，每个服务器拥有计算资源。任务可分割并按比例分配至不同无人机服务器，以减少计算延迟和平衡服务器负载，针对用户设备分配的任务，无人机边缘服务器对每个任务分配计算资源。

用户与服务器间传输速率由下式确定：



其中是带宽，是发射功率，是信道增益，是噪声功率。信道增益按照波长和距离计算。

服务器计算延迟按卸载比例、任务大小、CPU周期数和计算资源计算。优化目标是最小化计算延迟。优化问题表达式为：







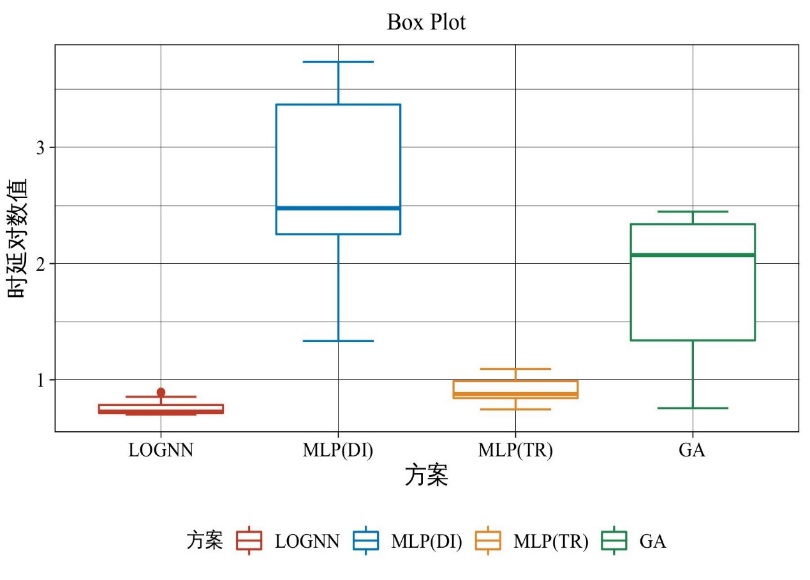








问题进一步建模为图链接权重回归问题，用户和服务器作为图节点，通信通道作为链接。用户节点特征矩阵和服务器节点特征矩阵定义了各自的任务大小和计算资源。双向图模型中，邻接特征数组定义了用户与服务器间的信道增益。优化变量被建模为链接权重，问题转化为最小化总延迟的加权和，同时满足资源分配约束。

通过图建模，采用1.3节中介绍的link-output GNN，并且为了降低训练成本，我们采用无监督学习进行模型的训练，损失函数值记为传输延迟和计算延迟。

### 2.2.2 仿真结果

为了证明所提出的基于GNN方案的有效性，这一部分我们设置了一个多无人机服务器多用户的场景，并随机生成存在多个不同数量的无人机边缘服务器和用户的场景，用户数量N始终是服务器数量M的两倍，并且与基于MLP的方案和基于GA的方案进行比较，其中MLP的方案包含预训练后直接推理的MLP方案记为MLP(DI)和即时训练的MLP方案记为MLP(TR)。

图 4 仿真结果

从箱型图中可以看出，MLP(DI)的体现出较差的性能，这是由于MLP缺乏可扩展性，采用直接使用固定大小的MLP并且参数不变，仅能为一部分边缘服务器和用户优化资源分配，其余的则随机分配。基于GA的方案作为一种近乎暴力搜索的优化方法，当优化问题复杂度较低且无人机和用户数量较小时，能够通过穷尽的进化搜索找到最优解。然而，随着问题规模的扩大，遗传算法难以精确定位目标函数的最优值。相比之下，LOGNN的箱型图明显低于其他三种方法，其中位数接近箱型图的下四分位数，这意味着至少有50%的数据点展示了极低的时延值。此外，整个箱体的位置也表明LOGNN在整体数据集中拥有最佳的时延性能。并且LOGNN的四分位间距非常小，说明其数据点间的差异较低，表明了其性能的一致性与稳定性，说明LOGNN能够为系统的预测性和可靠性提供保障。LGNN箱型数据点的最大值和最小值非常短，指出其数据的整体波动范围小，这也是高稳定性的一个重要标志。综上，LOGNN在时延和稳定性方面相较于其他方法表现出显著的优越性。这种性能上的优势使其在需要低时延和高稳定性的应用场景中，成为一个更加理想的选择。

## 结束语