A Survey of Graph-Based Resource Management in Wireless Networks - Part II: Learning Approaches

Yanpeng Dai, *Member, IEEE*, Ling Lyu, *Member, IEEE*, Nan Cheng, *Senior Member, IEEE*, Min Sheng, *Senior Member, IEEE*, Junyu Liu, *Member, IEEE*, Xiucheng Wang, Shuguang Cui, *Fellow, IEEE*, Lin Cai, *Fellow, IEEE*, Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—This two-part survey provides a comprehensive review of graph optimization and learning for resource management in wireless networks. In Part I, we introduced the fundamentals of graph optimization and provided a recent literature review of graph optimization for resource management in various wireless communication scenarios. In this part, we first present an overview of graph learning and introduce several modern graph neural network models. Then, a state-of-the-art literature review of graph learning for different resource management issues in wireless networks is provided, which covers power control, spectrum management, beamforming design, task scheduling, and aerial coverage planning. Furthermore, we discuss current technical challenges and future research directions of graph optimization and learning for resource management in future wireless networks.**

***Index Terms*—Wireless networks, resource management, graph**

I. Introduction

Graphs, as a classic discrete mathematical tool, have long been widely used for wireless network management, thanks to their capabilities to model complex relationships and interactions among wireless nodes. For resource management in wireless networks, there are two primary graph-based approaches at present which are graph optimization and graph learning. In Part I of this survey, we have introduced the fundamentals of graph optimization and provided a recent literature review of graph optimizaiton for resource management in wireless networks. Fig. 1 illustrates the organization of this two-part survey and the relationship between Part I and Part II. In this part, we shift the focus to graph learning and its application to resource management in wireless networks.

In recent years, graph-based resource managment is extending beyond graph optimization to involve graph learning, catering to the evolving demands in future wireless networks. There are increasing research outcomes to apply graph learning approaches to resource management in wireless networks, especially graph neural networks (GNNs) and graph embedding [1]–[9]. This is mainly because graph learning has the following advantages in resource management: 1) Scalability with network size, 2) Efficiency of training via wireless network data, 3) Generalization to dynamic network status, and 4) Compatibility with existing graph models of wireless networks. These advancements motivate us in this part of the survey to conduct a comprehensive literature review of graph learning for resource management in wireless network.

In this part, we focus on graph learning for resource management in wireless networks. Specifically, we present an overview of graph learning and introduce several modern GNN models in Section II. A state-of-the-art literature review of graph learning for resource management in wireless networks is provided in Section III, which is categorized by different issues including power control, spectrum management, beamforming design, task scheduling, and aerial coverage planning. The characteristics and components of each issue is demonstrated and the applicable graph learning approaches in the literature are comprehensively reviewed. Finally, we summarize technical challenges and future directions from both perspectives of graph optimization and learning in resource management. These challenges come from new features brought by the evolution of wireless networks. Future directions is aligned with the development of advanced graph optimization and learning techniques.

II. Fundamentals of Graph Learning

In this section, we first present an overview of graph learning and emphatically discuss GNNs that are a class of artificial neural networks capable of processing data and problems represented as graphs. Then, advanced GNN models and their extensions are introduced.

This paper was supported in part by the National Natural Science Foundation of China under Grant 62101089, in part by China Postdoctoral ScienceFoundation under Grant 2021M700655, in part by Doctoral Research-Startup Funds of Liaoning Province under Grant 2023-BS-068, in part by the Shenzhen Outstanding Talents Training Fund 202002, the Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, the Guangdong Provincial Key Laboratory of Future Networks of Intelligence under Grant No. 2022B1212010001, and the Shenzhen Key Laboratory of Big Data and Artificial Intelligence under Grant No. ZDSYS201707251409055.

Yanpeng Dai and Ling Lyu are with the School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China (Emails: yanpengdai@dlmu.edu.cn, linglyu@dlmu.edu.cn).

Nan Cheng, Min Sheng, Junyu Liu and Xiucheng Wang are with the State Key Laboratory of ISN and the School of Telecommunications Engineering, Xidian University, Xian 710071, China (e-mail: dr.nan.cheng@ieee.org, msheng@mail.xidian.edu.cn, junyuliu@xidian.edu.cn, xcwang_1@stu.xidian.edu.cn). Nan Cheng is the corresponding author.

Shuguang Cui is with the School of Science and Engineering (SSE), the Shenzhen Future Network of Intelligence Institute (FNii-Shenzhen), and the Guangdong Provincial Key Laboratory of Future Networks of Intelligence, the Chinese University of Hong Kong at Shenzhen, China, 518066 (e-mail: shuguangcui@cuhk.edu.cn).

Lin Cai is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada. (e-mail: cai@ece.uvic.ca).

Xuemin (Sherman) Shen is with the Department of Electrical, and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

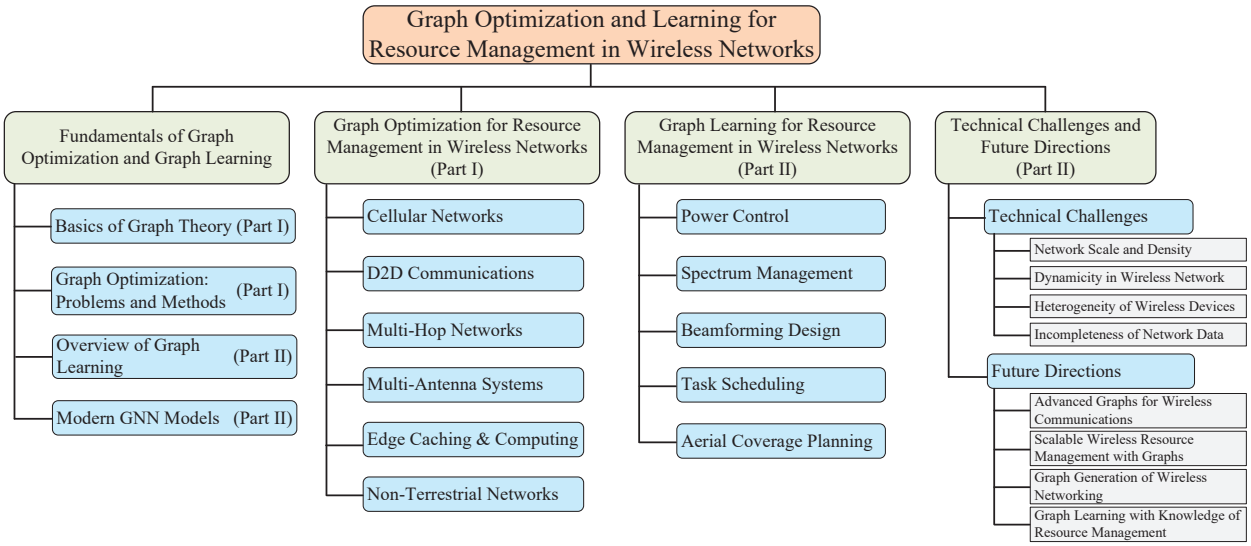


Figure 1. Organization of this paper and an overview of major topics.

A. Overview of Graph Learning

Due to the advance of big data and machine learning, graph learning, also known as graph representation learning, has made remarkable progress in the past several years. It aims to generate graph representation vectors to capture the structure and features of graphs accurately and efficiently. However, graph learning does not need to construct a graph strictly corresponding to a kind of graph optimization problems. It is necessary for graph learning to construct graph data consisting of the topological structure of a graph and the embeddings of vertices and edges. To be specific, graph embedding methods have been very successful in graph representation, which maps graph elements, e.g., vertices, edges, and subgraphs, to a lower dimensional space and preserves the properties of graphs. With the development of recent decade, GNNs have become a dominating category of graph embedding. A GNN can operate naturally on graph data to conduct learning tasks via extracting and utilizing features from the underlying graph.

1) Graph Embedding: Since the adjacency matrix is memory-consuming for representing large graphs, a graph $G = (\mathcal{V}, \mathcal{E})$ can be represented by their numerical features. For instance, a vertex in a graph can be represented with a set of features, e.g., weight, in-degree, our-degree, etc. However, it is difficult to find features that are important in different applications and can well represent whole properties of the graph. Therefore, graph embedding is proposed to deal with the issue of automatically generating representation vectors for graphs. Its main idea is formulating graph learning as a machine learning task in which the inputs are the elements and properties of graph and the outputs are embedding vectors. Graph embedding includes vertex, edge, and graph embeddings.

- *Vertex embedding:* It aims to learn a mapping function which transforms each vertex $v_m \in \mathcal{V}$ into an embedding vector of dimension d such that $d \ll \left|\mathcal{V}\right|$. The similarities between vertices in G are stored in the embedding space consisting of embedding vectors.
- *Edge embedding:* It is to encode each edge $e_n \in \mathcal{E}$ into an embedding vector of dimension d such that $d \ll \left|\mathcal{V}\right|$. The similarities between edges in G are stored in embedding vectors.
- *Subgraph embedding:* It transforms G or a subgraph H of G into an embedding vector of dimension d such that $d \ll \left|\mathcal{V}\right|$. Note that a graph embedding vector can be generated by aggregating embeddings of vertices and edges involved in G or H.

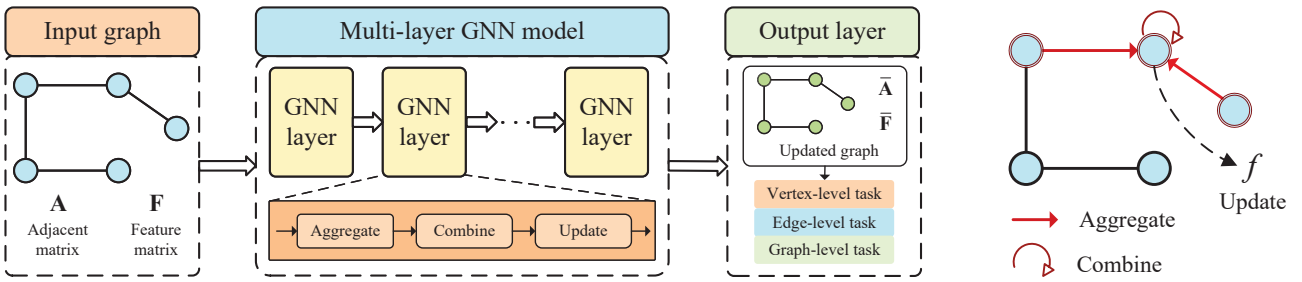
So far, almost existing graph embedding methods are vertex embedding, since it is the basis for edge embedding and subgraph embedding. The outputted embedding vectors can be used in different applications such as vertex classification, edge prediction, graph clustering, etc.

Traditional graph embedding methods are independent of GNNs and developed for static and dynamic graphs. For static graphs, there are matrix factorization method, random walk-based methods, and DL-based methods. For dynamic graphs, there are aggregation-based methods, random walk-based methods, DL-based methods, and temporal point process-based methods [10].

2) GNN: Modern GNNs are built upon the concept of message passing and adopt a graph-in graph-out architecture. The architecture of a GNN model in general consists of three parts that are the input graph, GNN layers, and the output layer, as shown in Fig. 2a.

The input graph $G = (\mathcal{V}, \mathcal{E})$ involves its own topological structure and all the features of vertices and edges in the graph. In practice, the topological structure is represented as an adjacency list that is derived from adjacency matrix. The n-th entry (i, j) in adjacency list describes the connectivity of e_n between v_i and v_j in G. The feature of each vertice or edge is represented as a d-dimension embedding vector. Therefore, we can obtain a vertex matrix with size $\left|\mathcal{V}\right| \times d^{\mathrm{v}}$ and an edge matrix with size $\left|\mathcal{V}\right| \times d^{\mathrm{e}}$, where d^{v} and d^{e} are independent.

GNN layers are the core part of entire model and dominate the model performance. A GNN layer uses message passing



(a) A general architecture of GNN.

(b) Example of message passing.

Figure 2. An illustration of GNN model.

to learn new embeddings aware of the graph's topological structure. Fig. 2b shows an example of how a GNN layer works. First, message passing enables each vertex to gather all the neighbours' embedding vectors, i.e., messages. Second, each vertex aggregates all received messages via an aggregate function, e.g., sum, and combines the aggregated result with its own message, e.g., concatenate. Then, all the aggregated and combined messages are passed through an update function that is usually an artificial neural network to output a new embedding vector for each vertex. Similar to vertices, message passing can also be applied to edges. Finally, a GNN layer outputs an updated graph with new embeddings, which maintains adjacency list and the sizes of vertex and edge matrices. We can stack several GNN layers together. With a GNN layer added, each vertex or edge can gather message one-hop further away. The message passing mechanism ensures two important properties of GNNs to graph data that are permutation invariance and permutation equivariance [11].

- *Permutation invariance:* GNNs do not depend on the arbitraty ordering of the rows or columns in adjacency matrix as well as vertex matrix and edge matrix. In other words, the output embedding vectors of GNN are invariant whatever any input matrix of graph data is permuted.
- *Permutation equivariance:* The output of GNN is permuted in a consistent way when the adjacency matrix or vertex/edge matrix is permuted.

GNNs are required to meet permutation invariance and/or equivariance in specific tasks.

The output layer is designed according to the aim of learning tasks. There are three kinds of GNN learning tasks that are vertex-level tasks, edge-level tasks, and graph-level tasks.

- *Vertex-level tasks:* They include vertex classification, vertex regression, vertex clustering, etc. The GNN model aims to output predicted labels or values for vertices. The output layer can be built by a multi-layer perceptron (MLP) or a softmax layer. GNN models for vertex-level tasks are usually permutation equivariant.
- *Edge-level tasks:* For edge classification and edge prediction, the GNN model aims to output the labels or connection probability for edges. Its output layer is similar to vertex-level tasks. If there are only vertex embeddings, the pooling module can be used to gather messages from vertices for outputting edge-related results. GNN models for edge-level tasks are almost permutation equivariant.
- *Graph-level tasks:* They include graph classification, graph regression, and graph matching. The GNN model aims to output a label or a graph embedding to an entire graph. The readout function and the pooling module can be used in the output layer that aggregates messages from vertices and edges to generate graph embedding vector. GNN models for graph-level tasks are usually permutation invariant.

Furthermore, GNN models can be integrated with reinforcement learning (RL) to deal with decision-making tasks in dynamic environments represented as dynamic graph data [12].

Many GNN models can be trained in a supervised, semi-supervised, or unsupervised manner with an end-to-end framework, according to learning tasks and label availability.

- *Supervised learning:* All the graph data for training is labeled. Graph classification is a typical supervised learning task.
- *Semi-supervised learning:* Partial graph data for training is labeled, while the rest is not. Most vertex and edge classifications tasks are typical semi-supervised learning.
- *Unsupervised learning:* All the graph data for training is unlabeled. Vertex regression, vertex clustering, and edge prediction are typical unsupervised learning tasks. Most resource management issues are currently solved by unsupervised learning with GNN.

At training phase, the results of the output layer is substituted into the loss function, e.g. cross-entropy and mean squared error (MSE), for optimizing model parameters. The loss function is designed according to learning tasks and training manners.

B. Modern GNN Models

As mentioned above, the message passing is the core of GNN design. Different message passing operators are key to differentiating GNN models. Convolution operator and recurrent operator are two basic message-passing operators in GNNs. On the basis of them, several advanced operators are developed. This subsection reviews several modern GNNs according to different message passing operators.

1) Recurrent GNNs: The most original recurrent GNNs (RecGNNs) use an identical module to recurrently process

graph data to extract vertex embeddings. Based on an information diffusion mechanism, this RecGNN model updates vertex states via neighbour information exchange until a stable equilibrium is reached [13]. Graph echo state network [14] and stochastic steady-state embedding [15] are developed to improve training efficiency and convergence performance.

The gating mechanism is exploited to design RecGNNs to diminish computational limitations. The gated GNN (GGNN) utilizes a gated recurrent units (GRU) as a recurrent module for message passing and uses back-propagation through time for training [16]. Give a graph $G = (\mathcal{V}, \mathcal{E})$, the hidden state $\mathbf{H}_m\,(t)$ of a vertex v_m at time step t is updated by its hidden state at time step $t-1$ along with those of its neighbours. Furthermore, the long short-term memory (LSTM) is also used to play the similar role as GRU for message passing on graph data [17].

2) Convolutional GNNs: The convolution operator is the most popular message passing operator for GNN models, which is usually categorized into spectral convolution and spatial convolution. Diverse convolutional GNNs (ConvGNNs) are based on these two operators.

Spectral convolution is first used in GNN models, due to its theoretical foundation in graph signal processing. Assume that a graph $G = (\mathcal{V}, \mathcal{E})$ is undirected. Let $\mathbf{x} \in \mathbf{R}^{|\mathcal{V}|}$ denote the vertex embedding of G, where x_m is the value of vertex v_m. Spectral graph convolution of $\mathbf{x}$ with a filter $\mathbf{g}_w \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is defined as $\mathbf{x} * \mathbf{g}_w = \mathbf{U}\mathbf{g}_w\mathbf{U}^{\mathrm{T}}\mathbf{x}$, where $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the matrix of eigenvectors of $\mathbf{L}$ with respect to the order of eigenvalues and $\mathbf{L}$ is normalized Laplacian matrix of G. The design of $\mathbf{g}_w$ differentiates spectral-based ConvGNNs.

- *Chebyshev spectral CNN:* The Chebyshev polynomials are used to approximate the filter $\mathbf{g}_w$ in Chebyshev spectral CNN (ChebNet) [18]. Define $T_k\,(\mathbf{x})$ as the Chebyshev polynomials of k-th order. The filter $\mathbf{g}_w$ can be approximated as $\mathbf{g}_w = \sum_{k=0}^{K} w_k T_k\left(\bar{\mathbf{\Lambda}}\right)$ where $\bar{\mathbf{\Lambda}} = 2\mathbf{\Lambda}/\lambda_{\max} - \mathbf{I}_{|\mathcal{V}|}$ and $\lambda_{\max}$ is the maximum eigenvalue in $\mathbf{\Lambda}$. $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{L}$ that are called the spectrum. Spectral convolution of $\mathbf{x}$ in ChebNet is expressed as $\mathbf{x} * \mathbf{g}_w = \mathbf{U}\big(\sum_{i=0}^{K} w_i T_i\big(\bar{\mathbf{\Lambda}}\big)\big)\mathbf{U}^{\mathrm{T}}\mathbf{x}$. All the w_i are learnable parameters in ChebNet.
- *Graph convolution network:* Let $\mathbf{A} = [a_{i,j}]$ denote the adjacency matrix of G and $\mathbf{D} = \mathrm{diag}\big(D_1, D_2, \ldots, D_{|\mathcal{V}|}\big)$ denote a diagonal matrix of vertex degrees. Graph convolution network (GCN) is a simplified version of ChebNet and takes the general form expressed as

$$\mathbf{H} = \mathbf{X} * \mathbf{g}_w = \bar{\mathbf{D}}^{-1/2}\bar{\mathbf{A}}\bar{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{W}, \tag{1}$$

  where $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{I}_{|\mathcal{V}|}$ and $\bar{\mathbf{D}} = \mathrm{diag}\big(\bar{D}_1, \ldots, \bar{D}_{|\mathcal{V}|}\big)$. $\mathbf{X} = \big[\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{V}|}\big]^{\mathrm{T}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ includes input vectors of vertex embedding with d dimensions. $\mathbf{W} \in \mathbb{R}^{d\times q}$ contains learnable parameters shared with all vertices. $\mathbf{H} = \big[\mathbf{h}_1, \ldots, \mathbf{h}_{|\mathcal{V}|}\big]^{\mathrm{T}} \in \mathbb{R}^{|\mathcal{V}|\times q}$ includes output vectors of vertex embedding with q dimensions.

Based on GCN, several variants are developed such as adaptive GCN [19] and graph wavelet neural network [20]. More importantly, GCN bridges spectral convolution and spatial convolution for graph data.

Spatial convolution utilizes the topological structure of a graph $G = (\mathcal{V}, \mathcal{E})$ to define the convolution operation. For instance, it convolves a vertex embedding with its neighbours' embeddings and repeats this operation for multiple steps to obtain the final embedding. *Message passing neural network* (MPNN) first proposes a general framework of spatial convolution over a graph [21]. In MPNN, the spatial convolution of vertex $v_m \in \mathcal{V}$ at step k is defined as a message passing function which is expressed as

$$\mathbf{h}_m^{(k)} = f_k\big(\mathbf{h}_m^{(k-1)}, \sum_{v_i\in\mathcal{N}_G(v_m)} g_k\big(\mathbf{h}_m^{(k-1)}, \mathbf{h}_i^{(k-1)}, \mathbf{e}_{i,m}\big)\big), \tag{2}$$

where $\mathbf{h}_m^{(k)}$ is the embedding of v_m at step k. $\mathbf{e}_{i,m}$ is the edge feature between v_i and v_m. $f_k\,(\cdot)$ and $g_k\,(\cdot)$ are functions with learnable parameters shared by all vertices at step k. Following the form of (2), the graph convolution performed by GCN as (1) can be rewritten as

$$\mathbf{h}_m^{(k)} = f_k\big(\mathbf{W}^{(k)} \sum_{v_i\in\mathcal{N}_m} \frac{\mathbf{h}_i^{(k-1)}}{\sqrt{\bar{D}_m\bar{D}_i}}\big), \tag{3}$$

where $\mathcal{N}_m = \mathcal{N}_G\,(v_m) \cup v_m$. (3) means that every vertex first aggregates all its neighbours' messages with its own message. Then, GCN substitutes aggregated message into a linear transformation $\mathbf{W}^{(k)} \in \mathbb{R}^{q\times d}$, whose result is finally inputted in $f_k\,(\cdot)$. MPNN can cover many existing GNN models by designating different $f_k\,(\cdot)$ and $g_k\,(\cdot)$ as well as pooling and readout functions.

- *Graph sample and aggregate:* If the graph contains vertices with a large number of neigbours, such as thousands or hundreds, it brings heavy computing load on MPNN for message aggregation. Instead of aggregating all neighbours' messages, graph sample and aggregate (GraphSAGE) [22] first uniformly samples a fixed number of neighbours for each vertex and then conduct message aggregation. GraphSAGE suggests three message aggregators, i.e., mean aggregator, LSTM aggregator, and pooling aggregator. For instance, GraphSAGE with mean aggregator is a variant of GCN that do sampling before convolution.
- *Graph attention network:* The attention mechanism is adopted for message passing in graph attention network (GAT) [23], which is shown to obtain better results than GraphSAGE in several specific tasks. GAT follows a self-attention strategy to learn the similarities between a vertex and its neighbours. Its graph convolution is expressed as

$$\mathbf{h}_m^{(k)} = f_k\big(\sum_{v_i\in\mathcal{N}_m} \alpha_{m,i}\mathbf{W}^{(k)}\mathbf{h}_i^{(k-1)}\big), \tag{4}$$

$$\alpha_{m,i} = g\big(h\big(\mathbf{a}\left[\mathbf{W}^{(k)}\mathbf{h}_i^{(k-1)} \parallel \mathbf{W}^{(k)}\mathbf{h}_i^{(k-1)}\right]\big)\big), \tag{5}$$

  where $g\,(\cdot)$ and $f\,(\cdot)$ are softmax function and LeakyReLU function, respectively. $\parallel$ is the concatenation operation. $\mathbf{a} \in \mathbb{R}^{1\times 2q}$ is a learnable parameter. GAT can be expanded to employ multi-head attention to stabilize the learning process of self-attention.
- *Graph isomorphism network:* It finds that GCN, GraphSAGE, and other MPNN-based models cannot learn

to distinguish certain simple topological structures of graphs. This limits the improvement of representational ability of GNNs. Hence, graph isomorphism network (GIN) [24] is proposed which performs the graph convolution as

$$\mathbf{h}_m^{(k)} = f_k\Big(\sum_{v_i \in \mathcal{N}_G(v_m)} \mathbf{h}_i^{(k-1)} + \big(1+\epsilon^k\big)\mathbf{h}_m^{(k-1)}\Big), \quad (6)$$

where the default setting of $f_k(\cdot)$ is an MLP. ϵ^k is a learnable parameter or a fixed scalar.

3) Spatial-temporal GNNs: In many real-world applications, graphs have time-varying features. In order to capture both spatial and temporal properties of a graph, spatial-temporal GNNs (STGNNs) are proposed, the majority of which follows two directions, i.e., recurrent neural network (RNN)-based and convolutional neural network (CNN)-based approaches.

RNN-based approaches integrate graph convolutional layers with RNNs. Graph convolutional layers are used to capture spatial relations among vertices and edges. RNNs are used to capture temporal relations between graphs over time. Given a graph $G = (\mathcal{V}, \mathcal{E})$, we suppose that a general RNN form is expressed as $\mathbf{H}(t) = f(\mathbf{W}\mathbf{X}(t) + \mathbf{Z}\mathbf{H}(t-1) + \mathbf{b})$, where $\mathbf{X}(t) \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the vector embedding matrix at time step t and $\mathbf{H}(t) \in \mathbb{R}^{|\mathcal{V}|\times q}$ is hidden state matrix at time step t. $\mathbf{W}$ and $\mathbf{Z}$ are learnable parameters. After integrating with graph convolution, it is rewritten as $\mathbf{H}(t) = f(g(\mathbf{X}(t), \mathbf{W}) + g(\mathbf{H}(t-1), \mathbf{Z}) + \mathbf{b})$, where $g(\cdot)$ is a graph convolutional layer. According to the selection of RNN module and $g(\cdot)$, different RNN-based STGNNs are proposed. For examples, graph convolutional recurrent network integrates LSTM with ChebNet [25]. A traffic flow prediction method combines GRU with GCN [26]. Structural-RNN presents an other thought to learn spatial-temporal graph data that purely uses a rich RNN mixture instead of mixing RNNs and graph convolutions [27].

CNN-based approaches interchangeably use graph convolutional layers and one dimensional (1D) CNN layers to learn spatial and temporal dependencies, respectively. Assume that $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times q\times T}$ is the input where T is the number of time steps. 1D CNN layer aggregates temporal information for each vertex along the time axis, i.e., $\mathbf{X}[m,:,:]$, while employing graph convolutional layer to aggregate spatial information at each time step, i.e., $\mathbf{X}[:,:,t]$. Spatial-temporal GCN (ST-GCN) composes a spatial-temporal convolutional block by stacking at least 1D CNN layers and a GCN layer [28], [29]. Graph WaveNet further proposes a self-adaptive adjacency matrix to perform graph convolutions without being given an adjacency matrix [30].

4) Graph autoencoders: The extension of auto-encoder to graph data has been a trend for generative model and unsupervised learning based on GNNs, which are called graph autoencoders (GAEs). Variational GAE (VGAE) first uses a GCN as the encoder and employs a simple inner product decoder [31]. The decoder aims to recover topological structure of a given graph. Specifically, the encoder outputs a hidden state $\mathbf{H}$ according to $\mathbf{X}$ and $\mathbf{A}$. Then, $\mathbf{H}$ is inputted into the decoder. The decoder outputs a reconstructed adjacency matrix $\hat{\mathbf{A}}$ according to $\mathbf{H}$. The training for GAE is to minimize the negative cross entropy between $\mathbf{A}$ and $\hat{\mathbf{A}}$. Instead of recovering the adjacency matrix, adversarially regularized VGAE employs generative adversarial network and GCN to construct GAE for robust vertex embeddings [32]. Based on VGAE, many GAE models are developed for graph embedding and generation [33].

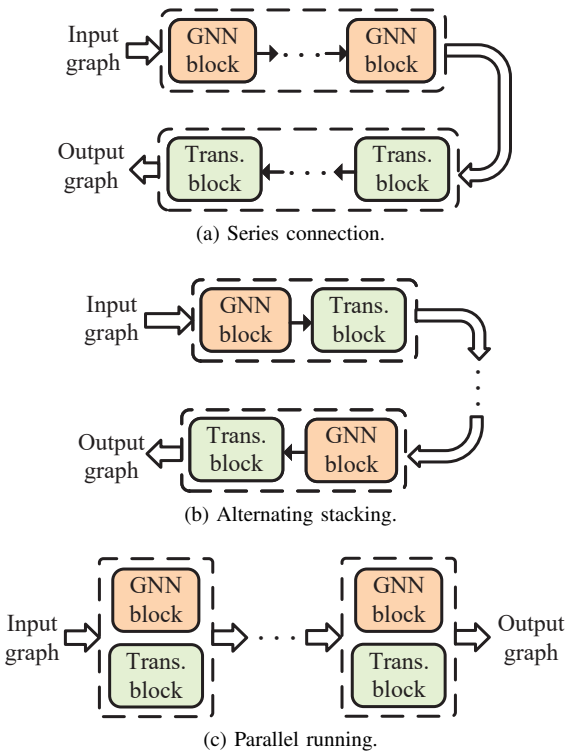


Figure 3. Combinations of GNNs and transformers.

5) Graph transformers: Transformer [34] is a generative model and has achieved great success in many artificial intelligence (AI) tasks and demonstrated great potential in modeling and processing graph data. To this end, graph transformers are designed for graph data and circumvent the shortcomings for GNNs, e.g., over-smoothing and over-squashing. Existing graph transformers are aware of topological structures by the following three primary means.

- *GNNs as auxiliary modules:* The most direct solution of involving structural knowledge into the transformer architecture is to combine GNNs as auxiliary modules with it. Generally, there are three combination types which are connecting GNN blocks and transformer blocks in series, stacking GNN blocks and transformer blocks alternately, or parallelizing GNN blocks and transformer blocks, as per Fig. 3. GraphTrans [35], Mesh Graphormer [36], and Graph-BERT [37] are three representatives of the above types, respectively.
- *Positional embeddings:* Topological structure of graph data can be compressed into positional embedding vectors that are further added to input features and then fed to the actual transformer model. This process is expressed as $\tilde{\mathbf{X}} = \mathbf{X} + f(\mathbf{P})$, where $\mathbf{X}$ is the matrix of input features. $\mathbf{P}$ contains positional embedding vectors and normally generated from the adjacent matrix. $f(\cdot)$ is a

function to align the dimension of vectors in $\mathbf{X}$ and $\mathbf{P}$. Laplacian eigenvectors and singular value decomposition of the adjacent matrix $\mathbf{A}$ can be used for positional embeddings in graph transformers [38], [39].

- *Attention bias:* The attention matrix can be modified by the bias matrix containing the information of topological structure. Let $\mathbf{Y}$ denote the attention matrix and $\mathbf{B}$ denote the bias matrix. The modified attention matrix $\tilde{\mathbf{Y}}$ is expressed as $\tilde{\mathbf{Y}} = f(\mathbf{Y}, \mathbf{B})$, where $f(\cdot)$ is the modified function. For example, Graphormer [40], as an attention bias-based graph transformer model, modifies the attention matrix as $\tilde{\mathbf{Y}} = \mathbf{Y} + \mathbf{B}$, where $\mathbf{B}$ includes learnable parameters about the shortest path distance between vertices.

Lessons learned 1: RecGNNs are expert in dealing with time-sequence graph data and capturing long-range dependencies among vertices or edges in graph. They can be applied to dynamic traffic prediction to enable proactive resource allocation in wireless networks. ConvGNNs are the most popular GNN architecture which capture local and global features of static graph via message passing and aggregation. They are applicable to static and quasi-static resource management issues in wireless networks. STGNNs are specialized for handling graph data with dynamic changes over time, which are usually designed by combining ConvGNNs with RNNs. They can be used for long-term resource management in dynamic wireless networks. GAEs are graph data oriented unsupervised learning and generative models, which consist of an encoder and a decoder. They can be applied to link scheduling, network deployment, and topology generation in large-scale wireless networks. Graph transformers are increasingly popular generative models for graph data, which utilize self-attention mechanism to capture global graph structure information and features. They are promised to tackle complex resource management issues in wireless networks.

III. Graph Learning for Resource Management in Wireless Networks

Compared to graph optimization, graph learning has the advantage in addressing large graphs and generalization ability, which has been applied to enhance resource management for all types of scenarios discussed in Part I of this survey [41]–[44]. This section focuses on five key issues and provides a literature review on the application of graph learning for resource management.

- *Power control:* Power control can determine desired signal strength at the receiver and energy consumption at the transmitter. It also plays a key role on interference coordination among transmission links. The communication and interference among transceivers can be represented via a graph. Thus, graph learning can be used as an effective tool for power control.
- *Spectrum management:* Spectrum resource is always precious and scarce for wireless communications and networking. User association, channel allocation/selection, and access control are typical spectrum management problems. Graph learning can rely on powerful representation capability to guarantee spectrum utilization.
- *Beamforming design:* Beamforming is a signal processing technique used in multi-antenna systems for directional signal transmission or reception. Efficient beamforming design plays an important role on increasing data-stream capacity between a transceiver pair. Graph learning has been proved to be effective for beamforming design.
- *Task scheduling:* Wireless networks have evolved from a data pipeline into a multi-task operating platform. Various wireless edge technologies are developed to utilize network resources to support communication, computation, caching, and learning tasks. Graph learning is employed for task scheduling at the edge of wireless networks.
- *Aerial Coverage Planning:* Unmanned aerial vehicles (UAVs) can act as aerial base stations (BSs) to enhance wireless coverage or aerial terminals covered by terrestrial BSs to realize various use cases. Therefore, aerial coverage planning is a new dimensionality to resource management for UAV communications, which can be handled by graph learning.

All the above five issues involve almost all essential resource management problems in various wireless networks. Furthermore, graph learning has been effectively applied to these five issues to improve resource utilization and compensate for the drawbacks of traditional approaches.

A. Power Control

Power control is one of the most representative resource management problems in wireless networks. The appropriate transmit power level can guarantee their desired signal strength at the receiver and save the energy at the transmitter, as well as coordinate interference among transmission links. At present, various GNN models are applied to power control in interference-limited networks and radio access networks (RANs).

1) Interference-Limited Networks: Several transmission links are distributed and each of them is a single-antenna transceiver pair consisting of one transmitter and one receiver. All the transmission links reuse the same spectrum bandwidth. It is necessary to carefully adjust the transmit power of each link to restrain interference due to spectrum reuse. Device-to-device (D2D) networks and multi-hop networks are typical interference-limited networks. Fig. 4 shows a general framework of GNNs for power control in interference-limited networks, which can be conducted in a centralized or decentralized manner.

In the centralized manner, the interference graph, or called conflict graph, is usually constructed. Each vertex represents a transmission link, whose weight is denoted by link weight or transmission quality. Each edge represents interference between any two transmission links, whose weight is denoted by interference intensity. MPNN-based models are directly implemented on the interference graph, where interference GCN (IGCNet) [3] and PCGNN [5] are two representatives. The results show that MPNN can achieve a higher convergence rate and better generalization than fully connected neural networks,

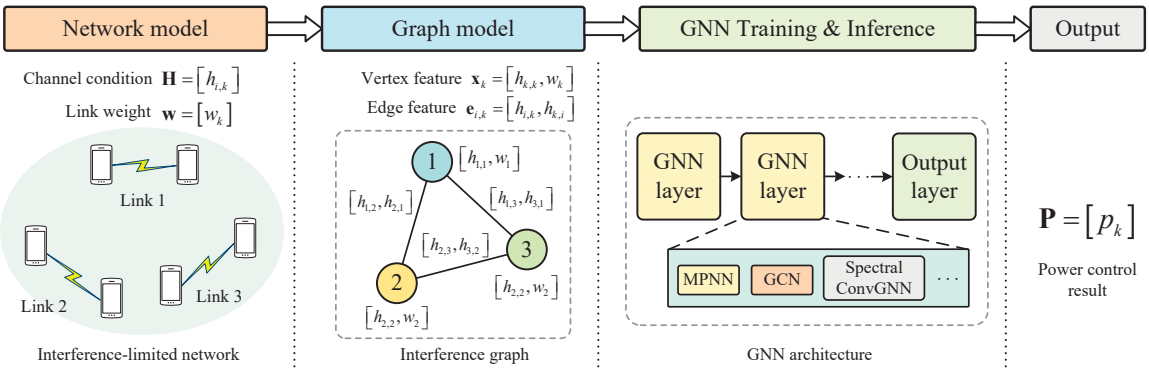


Figure 4. A general framework of GNN for power control in interference-limited networks.

such as MLP, particularly when the network scale is large [6]. In order to fit the randomly varying fading of wireless channels, the random edge graph neural network (REGNN) is proposed to adapt time-varying network topology. In REGNN, all hidden layers are ConvGNN layers and the edge weight is given by the randomly fading channel gain. In training phase of REGNN, a primal-dual learning method is designed to update the learning parameters and variables [1], [45]. Based on REGNN, the state-augmented algorithm is proposed which takes dual variables corresponding to network state as the GNN inputs to achieve the near-optimal performance and accelerate the convergence [46], [47]. To further accelerate the training and the inference of GNNs, an edge pruning method is designed for power control over the interference graph in D2D networks [48]. In addition, MPNNs are implemented to jointly optimize power control and communication direction in half-duplex interference-limited networks, e.g., Flex-Net [49], and design the power control algorithm in full-duplex D2D networks, e.g., F-GNN [50]. A STGNN model is designed for resource allocation in D2D underlaying cellular networks where GCNs extract spatial features and LSTM extracts temporal features for network traffic data to maximize the system throughput [51].

In the decentralized manner, since all the devices need to utilize local information to deduce global network states, local information exchange and aggregation among neighboring devices are crucial for the implementation of GNNs. The aggregation graph neural networks (Agg-GNN) is proposed to build local embeddings of global network state at each device via successive local state exchanges and aggregation on a graph of network topology. Then, the Agg-GNN based scheme employs ConvGNN layers to solve power control problems without explicit model knowledge [52], [53]. In training phase, the primal-dual learning method is performed in an asynchronous and model-free manner [54]. On the interference graph, modified REGNN architecture via meta-learning is designed for power control in a decentralized manner [55]. Results show that the integration of meta-learning and REGNN can effectively improve the iteration efficiency and facilitate fast adaptation to time-varying network topology [56]. The over-the-air computation technique is exploited to enable each devices to exchange and aggregate messages from its neighbours for local STGNN training and inference [57].

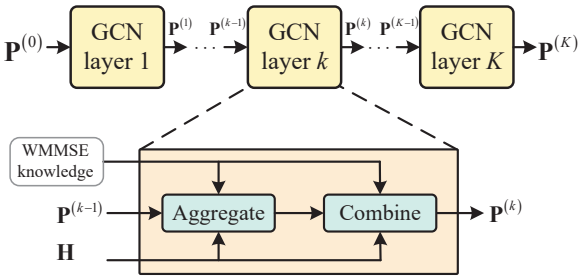


Figure 5. An example of algorithm unfolding with GNN.

A GAE model is proposed to realize decentralized power control in wireless ad hoc networks, which exhibits strong generalization to random network topologies [58].

Except for directly applying GNNs on graphs, the combination of GNNs and traditional optimization methods is effective for power control. Algorithm unfolding, or called unrolling, bridges GNNs and optimization methods [59]. For instance, the weighted minimum MSE method (WMMSE) is one of the most popular optimization algorithms for power control in interference-limited networks, which can be performed in a centralized or a decentralized manner. The unfolding WMMSE (UWMMSE) algorithm is developed to integrate GCN into the iteration process of WMMSE through algorithm unfolding [60], as per Fig. 5. Specifically, the iterations of WMMSE are unfolded as a sequential of GCN layers in which each layer uses message passing to update iteration parameters. It is shown that UWMMSE can achieve higher weighted sum-rate than WMMSE and pure GNN-based method with increased training time for convergence [61]. Building upon UWMMSE, the UWGNN algorithm follows a knowledge-driven principle that leverages the structure and domain knowledge of WMMSE to design the aggregation function and update function of GNN [62].

2) RANs: Multiple BSs, or called access points (APs), are deployed to provide radio access service in the coverage. All BSs reuse available spectrum resources entirely or partially. Hence, inter-cell interference coordination is required to boost network capacity and user experience, which can be realized by GNN-based power control. Currently, there are two frameworks of RANs that are multi-cell networks and cell-free networks.

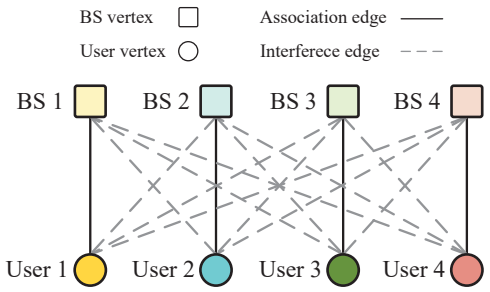


Figure 6. An example of heterogeneous graph.

In multi-cell networks, each BS acts as a centric node to associate with one or multiple users simultaneously while each user is associated with at most one BS. Since the nodes are with different roles and features, the heterogeneous graph is used to depict multi-cell networks. Different with homogeneous graph with only one type of vertices and edges, heterogeneous graphs are composed of different types of vertices or edges with different features, as per Fig. 6. In a heterogeneous bipartite graph, BSs and users can be respectively represented by two types of vertices with different features. Each edge represents a channel between one BS and one user where the edge weight is the corresponding channel gain. Heterogeneous GNNs (HetGNN) are the specialized GNN models employed on heterogeneous graphs. Results show that HetGNNs can efficiently optimize the power control policy for single-user-per-cell and multiple-user-per-cell scenarios with lower samples and training complexity than homogeneous GNNs and fully-connected neural networks [63], [64]. For the single-user-per-cell scenario, the interference graph can model the power control problem in multi-cell networks, on which various ConvGNN layers are designed [65], [66].

In cell-free networks, BS deployment is ultra-dense so that several BSs can form a cooperative serving cluster for each user. The association becomes user-centric and the cell edges become difficult to distinguish, known as cell-free. HetGNNs are still effective for the power control for cell-free networks over the bipartite heterogeneous graph which has two types of edges with different features. In each layer of this HetGNN, there are two types of messages carried by different edges that are the message from APs to users and the message from users to APs [5]. Furthermore, a directed heterogeneous graph is constructed for power control in cell-free networks. In this directed graph, each vertex is an AP-user pair whose feature is the channel gain between them and each edge is AP-type or user type to connect two vertices with the same AP or user. GAT is applied on this directed heterogeneous graph to produce the near-optimal power control [67].

Embedding GNNs into other optimization methods and learning paradigms is effective as well to solve the power control problem in multi-cell networks. Successive convex approximation (SCA) is an iterative optimization method which is popular for solving various scheduling problems in multi-cell networks. Based on the principle of algorithm unfolding, the unfolding SCA (USCA) algorithm is proposed to maximize the energy efficiency, which integrates GCNs into the iteration process of SCA to optimize power control for uplink multi-cell networks [68]. A stochastic gradient descent approach is used to progressively train GCN by the unsupervised loss and regularization terms [69]. Moreover, a GNN-based RL architecture is applied on REGNN for power control in wireless control systems that have the same framework with multi-cell networks. The proximal policy optimization (PPO) algorithm is used with the actor-critic networks parameterized by REGNNs [70].

Lessons learned 2: GNNs are effective for power control in large-scale and dense wireless networks, providing distinct advantages in both network performance and scalability. There are two primary implementations. First, ConvGNN models directly operate on graphs, e.g., interference graph, network topology graph, and heterogeneous graph. These models show superior generalizability and training efficiency than traditional DNN models. Second, algorithm unfolding is an effective bridge between GNNs and traditional optimization algorithms, such as WMMSE and SCA. It not only enhances optimization performance but also scales well to wireless networks of various sizes. Furthermore, GNN-based power control algorithms can be implemented by centralized training and execution or centralized training and decentralized execution. Table I summarizes the reviewed approaches using GNNs for power control along with references.

B. Spectrum Management

Due to the scarcity of spectrum resources, spectrum management is indispensable for wireless communications and networking. How to assign limited spectrum resources to transmission links or active users to increase network capacity and meet their requirements is the fundamental problem for spectrum management to address. We present the application of graph learning methods to spectrum management from the following three aspects, i.e., access control, user association, and channel allocation.

1) Access Control: Access control is to decide which part of transmission links or users can access the network at the current scheduling period. This process is also called link scheduling. Due to the limit of available spectrum resources and interference introduced by spectrum reuse, there is a ceiling to the number of transmission links or active users that cannot consistently be increased without control. Currently, graph learning methods, such as GNNs and graph embedding, conduct access control mainly by two graph models, i.e., the conflict graph and the maximum-weight independent set (MWIS).

On the conflict graph, REGNN is first implemented to realize access control by constraining transmit powers as binary variables in ad hoc networks where all the links reuse one spectrum channel. In this case, the access control problem is equivalent to a specific power control problem that is a binary selection of transmitting with $P_{\max}$ or not transmitting. Thus, REGNN for this access control problem can inherit its design for power control [2], [71]. In D2D networks, most of the literature define 0-1 variables as link access indicators

Table I
A SUMMARY OF APPROACHES USING GNN ARCHITECTURES FOR POWER CONTROL

Networks	References	Models	Categories	Learning Manners	Scopes
Interference-Limited Networks	[3], [5], [6]	MPNN	ConvGNN	Unsupervised	Centralized scheduling
	[1], [45]	Spectral ConvGNN	ConvGNN	Primal-dual unsupervised	
	[46], [47]	MPNN	ConvGNN	State-augmented Unsupervised	
	[48]–[50]	MPNN	ConvGNN	Unsupervised	
	[51]	GCN with LSTM	STGNN	Unsupervised	
	[52]–[54]	Spectral ConvGNN	ConvGNN	Primal-dual unsupervised	Decentralized scheduling
	[55], [56]	Spectral ConvGNN	ConvGNN	Meta-Learning	
	[58]	MPNN	GAE	Value-based RL	
	[57]	MPNN with RNN	STGNN	Unsupervised	Distributed scheduling
	[60], [61]	UWMMSE using GCN	GNN-based unfolding	Unsupervised	Centralized/decentralized scheduling
	[62]	UWGNN using MPNN	GNN-based unfolding	Unsupervised	
RANs	[63], [64]	HetGNN	ConvGNN	Unsupervised	Multi-cell network
	[65]	MPNN	ConvGNN	Unsupervised	
	[66]	MPNN	ConvGNN	Primal-dual unsupervised	
	[68], [69]	USCA using GCN	GNN-based unfolding	Unsupervised	
	[70]	MPNN with PPO	GNN-based RL	Actor-critic RL	
	[5]	HetGNN	ConvGNN	Unsupervised	Cell-free network
	[67]	GAT	ConvGNN	Unsupervised	

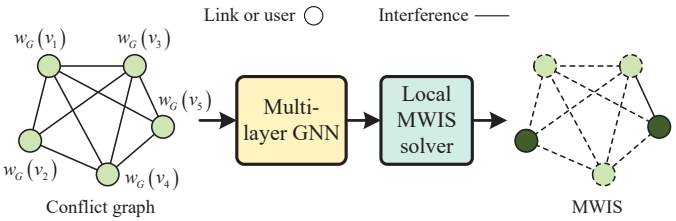


Figure 7. A general framework of GNN-based MWIS solver for access control.

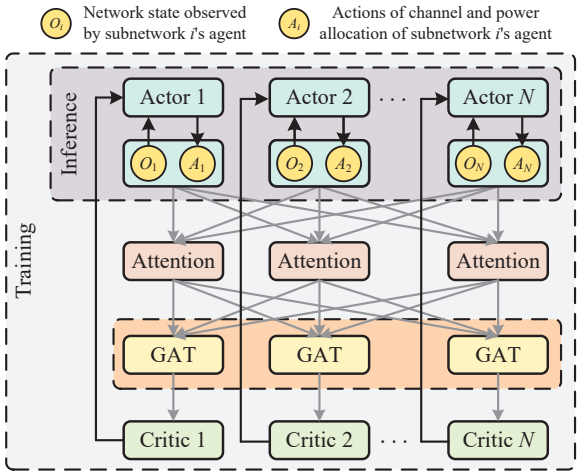


Figure 8. An example of GAT-integrated MARL for channel allocation.

and then optimize them via graph embedding techniques and GNNs. Specifically, the structure2vec, a deep learning (DL)-powered graph embedding technique, is introduced to access control for D2D links, which does not require accurate channel state information (CSI) and only need hundreds of training network layouts to achieve the near-optimal performance [8], [9]. GAE and MPNN are utilized to optimize 0-1 link access indicators in D2D networks via mapping the conflict graph to a low dimensional stochastic latent representation [72], [73]. Furthermore, MPNN is implemented to joint optimization of downlink user selection and power control in multi-cell networks [74]. A heterogeneous GAT-based multi-agent RL (MARL) algorithm is designed to optimize access control and resource allocation in a wireless networked control system [75].

The MWIS problem is formulated on a customized conflict graph with edge pruning. The graph only includes edges representing the interference stronger than a prescribed level and subtracts edges representing weaker interference. In this case, since the MWIS consists of transmission links or users allowed to access the networks, finding the MWIS is the process of access control. To this end, various GCN-based MWIS solvers are designed, as per Fig. 7. These solvers aim to maximize network throughput [76], reduce interference [77], or minimize average communication delay [78]. To further improve the efficiency of solving MWIS for access control, a fast centralized RL scheme based on GCN and deterministic policy gradient (DPG) are proposed, which are simultaneously applicable to the single-channel spectrum access and multi-channel spectrum access [7]. For multi-channel spectrum access, a two-stage topology-aware access control method is proposed which uses GCN to link scheduling and the attention mechanism to link prediction [79]. GAT is utilized to jointly optimize access and power control for massive connection in a ultra-dense wireless network [80].

2) Channel Allocation: In multi-channel wireless networks, because each user has diverse channel gains and suffers various levels of interference on different channels, channel allocation is required to assign appropriate channels to each user. The allocation indicators between channels and users are in general defined by 0-1 variables. At present, a variety of GNNs are applied to channel allocation in D2D communications and cellular networks.

For D2D communications, a MPNN-based model on the conflict graph is first designed for joint channel and power allocation that matches cellular uplink channels with D2D pairs for improving spectrum efficiency [73]. Furthermore, a GCN-based beam selection and link activation policy, called GBLinks, is proposed for millimeter-wave (mmWave) D2D networks, in which mmWave beams are available radio channels for D2D pairs [81]. Results show that GBLinks generalizes to varying network densities and network coverage region. The above two channel allocation methods for D2D communications are in a centralized manner. In a distributed manner, a GAT-based MARL approach is designed to realize spectrum channel allocation for multiple D2D pairs [82].

For cellular networks, a MPNN-based model is proposed for optimizing transmit power and spectrum allocation in single-cell networks, which is shown to generalize well to the number of users [83]. A MARL algorithm combining GCN and deep Q-network (DQN) is proposed for channel allocation in HetNets consisting of macro BSs and small BSs. In this work, small BSs exploit non-orthogonal multiple access (NOMA) and cognitive radio (CR) to reuse spectrum channels of macro BSs to expand network capacity [84]. GAT is employed to extract potential interference relationships among subnetworks which is modeled as a complete graph. Then, multiple GAT models are integrated with multi-agent actor-critic architecture for channel and power allocation to improve channel utilization [85], as Fig. 8 shows. In addition, MPNN is used with the policy gradient method to solve the channel allocation problem in wireless local area networks (WLANs) which are modeled as multi-cell networks with random access scheme [86]. GAT is utilized to design a STGNN model to proactively allocate spectrum bandwidth in multi-cell vehicle-to-everything networks [87].

3) User Association: In wireless networks including multiple BSs or APs, how to associate each user with one serving BS or AP is an essential issue to be investigated, i.e., user association. Traditional user association methods follow simple principles such as the nearest distance and the maximum receive signal strength to enhance link quality and network capacity. Due to key performance indicators varying with the wireless network evolution, the aim of user association has gradually expanded from network capacity to load balancing, energy efficiency, user fairness, etc. Recently, graph learning demonstrates the potential on designing efficient and scalable user association schemes.

GraphSAGE is utilized to design a user association and power allocation scheme based on unsupervised graph learning in ultra-dense HetNets [88]. Double DQN (DDQN) is further combined with GCN to design a user association algorithm, named GROWS, to maximize network throughput with user fairness in small-cell networks [89]. A variant of GAT model is proposed to jointly optimize user association and beam selection in mmWave-integrated HetNets to maximize the system sum-rate [90]. To facilitate intelligent management for terahertz multi-cell networks, a GNN-based digital twin architecture is proposed where MPNN is used to jointly optimize user association and power allocation to maximize average rate of users [91]. In cell-free massive multiple-input multiple-output (MIMO) systems, GraphSAGE leverages the path loss and shadow fading to predict the potential association between APs and users. It simultaneously works on a homogeneous graph including only AP nodes and a heterogeneous graph including AP nodes and user nodes [92]. Catering to the delay requirement of uRLLC traffic, an edge-wise gated GNN (EG-GNN) is developed for user association in a mobile edge computing (MEC) network. The EG-GNN updates node features and gates in an iterative manner, in which each gate is designated by a fully connected neural network at the uRLLC device [93]. In a vehicular edge computing network, a task scheduling method based on spatial ConvGNNs and PPO is proposed to improve delay performance by optimizing the association between tasks and edge servers [94]. In addition, a GCN-based network access scheme is proposed for LTE-WLAN heterogeneous networks that select the AP with the best performance for each user among different types of APs [95], [96].

Lessons learned 3: Graph learning techniques are compatible with various spectrum management issues to increase spectrum utilization and lower management overhead. Because most spectrum management problems belong to the field of 0-1 integer programming, traditional methods usually face high computational complexity and their performance heavily depends on CSI accuracy. Due to powerful generalizability, GNN and graph embedding methods do not require accurate CSI and just need fewer network samples to achieve near-optimal results of spectrum management. Meanwhile, these methods scale well to the number of users and the number of spectrum resources. Conflict graph, binary graph, and MWIS graph are popular graphs for GNNs and graph embedding techniques for spectrum management. Furthermore, GNN-based RL is an effective approach for spectrum management. On the one hand, RL algorithms can be used to train GNN models to enhance solving quality and performance for 0-1 integer programming problems. On the other hand, GNN models are used to extract network state information to improve the decision making of RL in a dynamic environment. Table II summarizes the reviewed approaches using graph learning for spectrum management along with references.

C. Beamforming Design

Beamforming is a signal processing technique to focus transmitted signals to specific direction in multi-antenna systems such as MIMO and multiple-input single-output (MISO). A beamformer changes the transmitting directionality of the antenna array via controlling the phase and relative amplitude of the signal of each antenna in the array. This process is called beamforming design that directly determines beam direction and achievable transmission capacity. Graph learning has been implemented into beamforming design for MIMO interference channel, multi-user MIMO (MU-MIMO) systems, and reconfigurable intelligent surface (RIS)-aided MIMO systems.

1) MIMO Interference Channel : Multiple transceiver pairs, like D2D pairs, occupy the same radio channel. Each of them is equipped with multiple antennas in the transmitter and/or

Table II
A SUMMARY OF APPROACHES USING GRAPH LEARNING FOR SPECTRUM MANAGEMENT

Issues	References	Models	Categories	Learning Manners	Scopes
Access Control	[2], [71]	MPNN	ConvGNN	Primal-dual unsupervised	Ad hoc network
	[76]	GCN	ConvGNN	Unsupervised	
	[8], [9]	Structure2vec	Graph embedding	Supervised	D2D
	[72]	GCN	GAE	Unsupervised	
	[73]	MPNN	ConvGNN	Supervised	
	[80]	GAT	ConvGNN	Unsupervised	
	[74]	MPNN	ConvGNN	Primal-dual unsupervised	Multi-cell network
	[75]	GAT	ConvGNN	Actor-critic RL	
	[77]	GCN	ConvGNN	Unsupervised	Multi-hop network
	[78]	GCN	ConvGNN	Value-based RL	
	[7]	GCN with DPG	GNN-based RL	Policy-based RL	
	[79]	GCN	ConvGNN	Supervised	
Channel Allocation	[73]	MPNN	ConvGNN	Supervised	D2D underlaying cellular network
	[82]	GAT based MARL	GNN-based RL	Actor-critic RL	
	[81]	GCN	ConvGNN	Primal-dual unsupervised	D2D
	[84]	GCN based MARL	GNN-based RL	Value-based RL	HetNet
	[83]	MPNN	ConvGNN	Unsupervised	Single-cell network
	[85]	GAT based MARL	GNN-based RL	Soft actor-critic RL	Multi-cell network
	[87]	GAT with GRU	STGNN	Supervised	
	[86]	MPNN	ConvGNN	Policy gradient	WLAN
User Association	[88]	GraphSAGE	ConvGNN	Unsupervised	HetNet
	[90]	GAT	ConvGNN	Primal-dual unsupervised	
	[89]	GCN based DDQN	GNN-based RL	Value-based RL	Multi-cell network
	[91]	MPNN	ConvGNN	Unsupervised	
	[92]	GraphSAGE	ConvGNN	Supervised	Cell-free network
	[93]	MPNN	ConvGNN	Unsupervised	MEC
	[94]	GAT or GraphSAGE based PPO	GNN-based RL	Policy-based RL	
	[95], [96]	GCN	ConvGNN	Supervised	LTE-WLAN

the receiver. In order to coordinate interference caused by channel reuse, the beamforming of each transceiver pair should be carefully designed to match channel gain and transmission direction. The beamforming design in interference channel is similar to the power control in interference-limited networks. The main difference between them is that the scalar denoting transmit power becomes a vector denoting the beamformer. In fact, the beamforming design is equivalent to a power control problem for single-antenna systems. Therefore, the interference graph is still efficient to model the beamforming design problem [5].

To tackle the above beamforming design problem, the WCGCN is proposed which is based on MPNN. In this work, MPNN-based beamforming algorithm is proved to be equivalent to distributed local optimization algorithm in terms of performance and applicability [4]. HIGNN is the other MPNN-based algorithm for beamforming design in MISO interference channel. Results show that HIGNN is scalable to wireless networks of various sizes after trained on small-scale networks [97]. An edge-update mechanism is designed to propose an edge-node GNN (ENGNN) which can further efficiently deal with both edge and node variables for beamforming design with low computation cost and enhanced generalization [98]. A meta-gating GNN model is proposed for beamforming design, which exploits WCGCN to design an inner GNN and an outer GNN. In this framework, meta learning algorithm is used in training process and element-wise gating operation is used to multiply the outputs of inner and outer GNNs [99]. GNN-based beamforming design algorithms have also been further extended from MISO to MIMO interference channel [100]. Furthermore, UWMMSE algorithm is implemented to the beamforming design in MIMO interference channel for the sum-rate maximization, which is shown to be superior in performance, generalizability, and robustness [101]. A MPNN-based distributed beamforming and power control algorithm is designed in a massive ultra-reliability and low-latency communication (uRLLC) network to reduce signaling overhead and computation delay [102]. An edge enhanced GAT model is used to optimize beamforming design in MISO interference channel with statistical CSI [103].

2) MU-MIMO Systems: MU-MIMO allows a BS to serve multiple users on one frequency/time resource by means of multi-antenna techniques. Its main idea is to form a dedicated beam for each user, enabling the transmission of independent data streams to different users. Hence, one-to-multiple beamforming dominates the performance of MU-MIMO systems which is different from one-to-one beamforming. At present, graph learning-based beamforming design has shown the effectiveness on inter-user interference coordination and network capacity enhancement in single-cell and multi-cell MU-MIMO systems.

In single-cell MU-MIMO, a bipartite GNN (BGNN) framework is developed for beamforming design. In BGNN, message passing is realized by GCNs over a bipartite graph, as Fig. 9 shows. There are two disjoint vertex sets representing antennas and users, respectively, with different types of features. The edges between them are featured by channel gains [104]. The spectral graph convolution is combined with algorithm unfolding to propose a USRMNet algorithm for beamforming design in a uRLLC system. Results show that the USRMNet can efficiently maximize the sum rate with

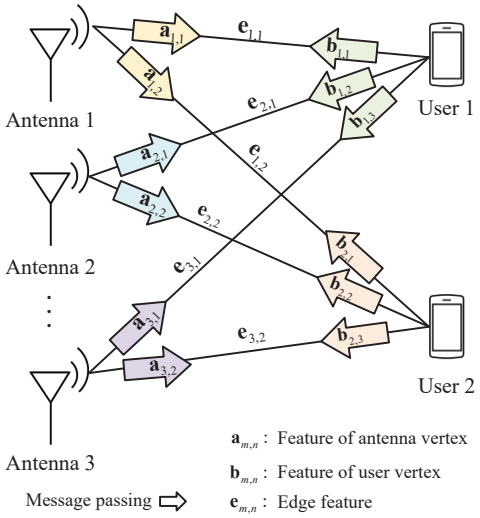


Figure 9. An example of message passing over a bipartite graph for MU-MIMO beamforming design.

the constraint of desirable decoding error probability [105]. GAT-based beamforming design models are proposed for energy efficiency maximization, sum-rate maximization, max-min user fairness, and sum secure rate maximization in single-cell MU-MIMO systems [106]–[109]. Moreover, graph learning methods are used in joint optimization of beamforming design and other aspects of single-cell MU-MIMO systems. A GCN-based learning algorithm is proposed to jointly optimize user scheduling and beamforming design for maximizing the number of scheduled users [110]. A graph neural imitation learning method is designed to jointly optimize beamforming design and antenna selection for saving transmit powers, where imitation learning accelerates the training of GNN model [111], [112]. A unicast-multicast GNN (UMGNN) architecture is proposed to jointly design multicast and unicast beamformers with imperfect CSI [113].

In multi-cell MU-MIMO, the beamforming evolves into coordinated beamforming among multiple BSs. Inter-cell interference is an important feature for GNNs to learn in this scenario. The ENGNN is implemented to optimize beam powers and coordinated beamforming design, which shows low computational cost and generalization [98]. A GCN-WMMSE architecture is developed which utilizes deep unfolding in coordinated downlink beamforming. Results show that this architecture significantly reduces the number of required iterations to achieve similar performance to WMMSE [114]. Furthermore, a self-optimized MPNN, namely AutoGNN, is proposed to learn coordinated beamforming in a distributed manner meanwhile automatically learning the depths of desirable GNN layers and the sizes of feature embedding [115].

3) RIS-Aided MIMO Systems: RIS is a programmable surface structure with the capability of enhancing wireless propagation environment. It consists of a large number of tunable reflective elements to intelligently reflect electromagnetic waves via adjusting phase shifts of reflective elements. In existing works, RIS is usually deployed in MIMO systems to focus the beam toward the directions of users, as per

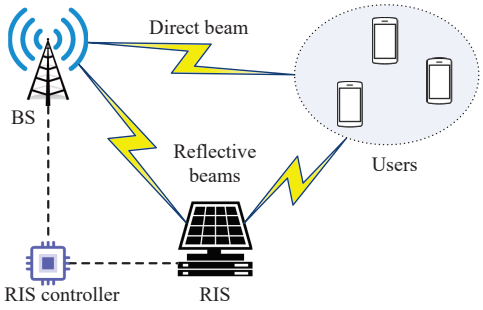


Figure 10. An example of RIS-aided MIMO systems.

Fig. 10, which relies on the optimization of beamformer and phase shifts simultaneously. Graph learning methods have been applied to the optimization of RIS-aided MIMO systems.

Graph learning is initially used to optimize single-cell MU-MIMO with the aid of a single RIS. A MPNN-based model is developed to map the uplink pilots to the beamformers at the BS and the phase shifts at the RIS. This architecture shows its generalizability to the number of users because it is permutation invariant for phase shifts and permutation equivariant for beamformers [116]. Based on this advance, a dual-GNN framework is proposed in which two MPNNs respectively optimize user scheduling and RIS configuration and then are used to fine-tune the beamformers [117], [118]. Similarly, a HetGNN framework consisting of two GNNs are designed where one is used to learn channel estimation and the other is used to learn beamformers and phase shifts [119]. A multi-head GAT is exploited to optimize BS precoding and RIS phase shift to maximize the sum rate of users [120]. A heterogeneous graph transformer model is employed in a RIS-aided MU-MIMO system to optimize the precoding, phase shift, and bandwidth allocation [121].

Graph learning is further employed to optimize multi-RIS aided MIMO systems. A novel MARL integrated with WCGCN, called GE-VDAC, is developed to reconfigure hybrid spatial beamforming in a multi-AP multi-RIS MIMO-NOMA network [122]. A HetGNN model is designed to jointly optimize beamformers, phase shifts, and user-RIS association. Results show that this architecture efficiently improves the quality of service for users compared to fully association between RISs and users [123]. A MPNN model is proposed to jointly optimize beamforming pattern, rate allocation, and phase shift in a multi-RIS aided downlink MU-MISO system with rate splitting multiple access [124].

Lessons learned 4: GNNs provide an efficient and alternative solution for beamforming design in multi-antenna systems. ConvGNN and algorithm unfolding with GNN are effective for designing beamforming optimization algorithms, similar to power control. Several existing GNN models for power control, such as WCGCN and UWMMSE, are also applicable to beamforming design. The literature shows that the performance and applicability of MPNN are equivalent to distributed local beamforming optimization algorithms. Moreover, unfolding traditional beamforming optimization algorithms via GNN shows the potential to outperform original algorithms in

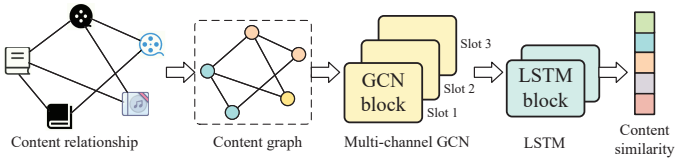


Figure 11. An illustration of STGNN for content similarity representation in edge caching.

terms of performance, generalizability, and robustness. These evidences facilitate the application of GNNs in various massive MIMO systems. Recently, GNN-based methods have been employed in the optimization of RIS-aided MIMO systems which demonstrate the efficiency of jointly optimizing beamforming and phase shifts, as well as the generalizability to the system scale. Table III summarizes the reviewed approaches using graph learning for beamforming design along with references.

D. Task Scheduling

Wireless networks are becoming multi-task operating platforms to utilize multi-dimensional resources to satisfy various service requirements. To this end, diverse wireless edge technologies, such as edge caching, MEC, integrated sensing and communication (ISAC), and edge learning, are developed to enable different network elements to cooperate with each other and perform tasks of communication, computation, caching, learning, and sensing. GNNs have been used for task scheduling at aspects of content caching, computation offloading, and edge learning.

1) Content Caching: Deciding where to cache the content and which content to cache are typical tasks addressed by GNNs in edge caching systems. GAT is used to propose a fully decentralized soft MARL algorithm for cooperative caching and fetching in D2D networks. In this work, GAT accelerates the coordination of content caching among D2D pairs [125]. Furthermore, a user preference learning-based proactive edge caching strategy is developed which utilizes STGNN to learn content similarity. In this work, STGNN consists of multi-channel GCN and LSTM models, as Fig. 11 shows. GCN extracts spatial relations among contents over the content graph in a time slot and then LSTM captures temporal correlations among contents in different time slots to output content similarity representation [126].

2) Computation Offloading: GNNs can be applied to computation offloading for interdependent tasks and independent tasks. In computation offloading of interdependent tasks, directed acyclic graph (DAG) is usually used to model the order of execution and the interplay of different tasks. Several GNN-based computation offloading schemes are developed by means of DAGs. GCN-embedded RL schemes are proposed to optimize task offloading between one user and one BS where all the tasks are generated through partitioning one application and hence interdependent with each other [127], [128]. The GASTO, a meta-RL computation algorithm based GAT and LSTM, is developed to cope with the dynamic and uncertainty of MEC environment which is modeled by a specialized DAG [129]. Furthermore, GNN-augmented RL schemes are developed for interdependent task offloading in multi-BS MEC systems [130], [131]. In addition, a GCN model over a conflict graph is proposed to optimize task offloading in a multi-cell MEC network, in which each vertex represents a task and each edge connects two tasks with the conflict on execution time [132].

In computation offloading of independent tasks, GNNs work on the graph of network topology where vertices represent network elements, e.g., BSs, users, and tasks. Edges represent the association among these elements. Due to network topology changes and device collaboration, GNN-based RL approaches are employed to optimize offloading decision and resource allocation for independent tasks [133], [134]. To be specific, a GCN-based actor-critic network is proposed to learn offloading decision for a single-cell MEC network [135].

For a multi-cell MEC network, a GAT-integrated DQN is proposed to optimize offloading decision and resource allocation [136]. A novel link-output GNN-based resource management scheme is proposed for partial offloading in a multi-cell MEC network [137]. An inverse RL algorithm is proposed which utilizes MPNN and GRU to dynamically optimize offloading decision and computation/communication resource allocation in a cloud-edge-end computing network [138]. To realize cooperative task offloading and service caching, a GAT-based deep recurrent Q-network (DRQN) is devised for a digital twin-empowered multi-cell MEC network to improve the quality-of-service of the overall system [139]. A multi-cluster cooperative offloading scheme based on GCN assisted MARL is proposed for virtual reality tasks in a MEC network [140]. In addition to the above works, MPNN is implemented to design a novel resource slicing strategy in MEC networks [141].

3) ISAC: ISAC is emerging as a key feature of future wireless networks, allowing for the exploitation of current network resources and infrastructures to build a perceptive network. Due the coexistence of sensing and communication tasks, spectrum sharing and link scheduling between them are primary problems for resource management. GAT is used to propose an intelligent resource allocation algorithm for simultaneously meeting the requirements of communication and sensing tasks in an ISAC system [142]. A MPNN-based model is designed to optimize link scheduling for an ISAC-enabled vehicular network to maximize sum rate while meeting sensing requirements, where all the vehicles reuse the same spectrum bandwidth and can operate in either communication mode or sensing mode [143]. Based on [143], a dynamic GNN approach is proposed to select appropriate message pasing functions according to vehicle network topology [144].

4) Edge Learning: AI is rapidly merging together with wireless communications and networking. Edge learning is one of its representative instances, which enables wireless edge facilities, e.g., BSs, APs, and user devices, to execute machine learning tasks such as model training and inference. GNNs have been utilized to task scheduling for distributed edge computing and federated learning (FL) in wireless networks.

In a distributed edge computing system, there are several DL tasks required to offload to diverse Internet-of-Things (IoT) devices. To tackle this problem, a GCN-based task allocation

Table III
A SUMMARY OF APPROACHES USING GRAPH LEARNING FOR BEAMFORMING DESIGN

Systems	References	Models	Categories	Learning Manners	Scopes
MIMO Interference Channel	[4]	MPNN	ConvGNN	Unsupervised	D2D
	[97]	HetGNN	ConvGNN	Unsupervised	
	[98], [102]	MPNN	ConvGNN	Unsupervised	
	[99]	MPNN	ConvGNN	Meta-learning	
	[100]	HetGNN	ConvGNN	Unsupervised	
	[103]	GAT	ConvGNN	Unsupervised	
	[101]	UWMMSE using GCN	GNN-based unfolding	Unsupervised	Ad hoc network
MU-MIMO	[104], [107]	GCN	ConvGNN	Unsupervised	Single-cell network
	[105]	Algorithm unfolding using spectral ConvGNN	GNN-based unfolding	Primal-dual unsupervised	
	[106]	GAT	ConvGNN	Supervised	
	[108], [109]	GAT	ConvGNN	Unsupervised	
	[110]	GCN	ConvGNN	Unsupervised	
	[111], [112]	MPNN	ConvGNN	Imitation learning	
	[113]	MPNN	ConvGNN	Unsupervised	
	[98]	MPNN	ConvGNN	Unsupervised	Multi-cell network
	[114]	Algorithm unfolding using GCN	GNN-based unfolding	Unsupervised	
	[115]	MPNN with auto-learning	ConvGNN	Unsupervised	
RIS-Aided MIMO	[116]–[118]	MPNN	ConvGNN	Unsupervised	Single-RIS system
	[119]	HetGNN	ConvGNN	Unsupervised	
	[120]	GAT	ConvGNN	Unsupervised	
	[121]	Heterogeneous graph transformer	Graph transformer	Unsupervised	
	[122]	MARL with MPNN	GNN-based RL	Actor-critic RL	Multi-RIS system
	[123]	HetGNN	ConvGNN	Supervised	
	[124]	MPNN	ConvGNN	Unsupervised	

algorithm is proposed which works on a heterogeneous graph and learns the optimal matching between the workloads of DL tasks and available resources of IoT devices [145]. In FL over wireless networks, BS is equipped with or connected to the FL server. User devices are FL participants. Each user device owns a private dataset used to train a local model and send local model parameters to the FL server. Then, the FL server aggregates all the collected local model parameters to generate a global model, as per Fig. 12. In order to improve the offloading efficiency of local model parameters, a GCN-based power allocation policy is devised which uses the primal-dual algorithm to solve considered optimization problem and outperforms MLP in transmission success rate and FL global performance [146]. For a RIS-aided FL system, a novel MPNN-based learning algorithm is proposed to optimize transmit powers, denoising factor, and phase shifts [147]. Furthermore, GNNs are employed for FL-based applications in wireless networks. In a FL-based aerial-ground air quality sensing framework, a GCN and LSTM based STGNN is used to achieve accurate and real-time air quality inference [148]. In wireless FL service market, an automated strategy-proof mechanism based DQN and GCN is developed to improve social welfare [149].

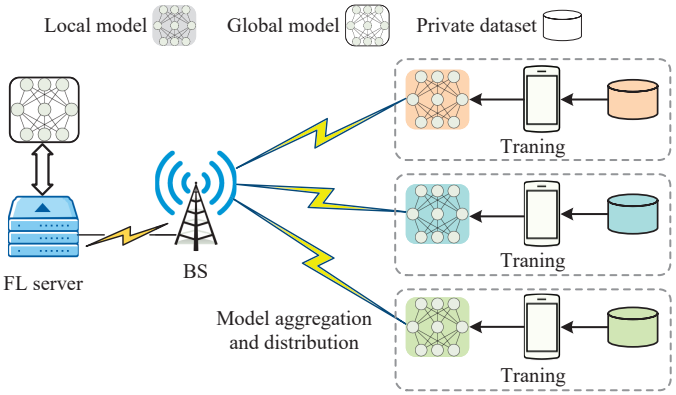


Figure 12. An illustration of FL.

Lessons learned 5: Various GNN models and learning approaches are applicable for utilizing multi-dimensional resources for task scheduling at terms of communication, computing, caching, sensing and learning. There are three popular GNN-based approaches for task scheduling. First, ConvGNN can extract implicit information from the graph derived from a specific task, e.g., content similarity, task interdependency, association opportunity, and interference distribution. This information further contributes to the conduction of task scheduling. Second, STGNN is able to capture not only spatial features from underlying graph but also temporal features from time-sequential data, e.g., historical data of content request and time-varying network topology. These features are exploitable to time-sensitive task scheduling. Third, GNN-based RL is suitable for designing the task scheduling scheme in a dynamic wireless environment and cooperative task scheduling among devices. The key lies in the choice of GNN models and RL algorithms. In addition, conflict graph and network topology graph are commonly used for GNN-based task scheduling. Table IV summarizes the reviewed approaches using GNN

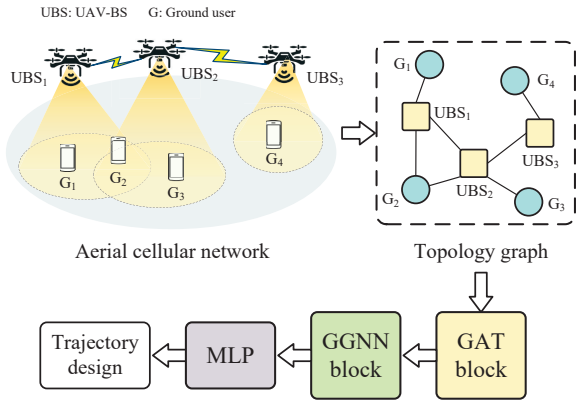


Figure 13. An illustration of GAT and GGNN for trajectory design in aerial cellular networks.

architectures for task scheduling along with references.

E. Aerial Coverage Planning

UAV has great potentials to enhance the connectivity and broaden the applications of wireless networks. On the one hand, wireless networks can expand the coverage by exploiting UAVs as aerial BSs and relays. On the other hand, UAVs can be aerial terminals covered by terrestrial BSs to enable various use cases such as smart cities, disaster reconstruction, intelligent transportation, etc. The mobility of UAVs leads to flexible network topology via controlling flying trajectory and hovering position, which make aerial coverage planning be a new dimensionality as well as challenge to resource management. Due to excellent generalization and scalability, GNNs are utilized to carry out aerial coverage planning.

A GCN-based trajectory planning algorithm is developed for UAV swarm communication which rebuilds the connectivity among UAVs in the swarm during self-healing process. This algorithm is also combined with meta learning to facilitate the off-line learning and the on-line execution of GCN [150]. In aerial cellular networks, GAT and GGNN are employed for cooperative trajectory design of multiple UAV-BSs to improve long-term system throughput [151], as per Fig. 13. GAT is further used to obtain extra spatial information via inter-UAV communications to plan flying paths of multiple UAV-BSs for optimal network coverage under partial observation [152]. A GAT-based MARL algorithm is developed for trajectory planning and resource assignment to ensure load balance among UAV-BSs [153]. A GNN-empowered partial observation MARL method is proposed for distributed trajectory planning to optimize the age of information of users [154]. In a UAV-relaying IoT network, a dual-stage GNN optimization framework is proposed in which a GCN-LSTM model is used to select the best relay path and then a GAT-LSTM model is used to optimize the deployment of UAV relays. Results show that this framework can achieve comparable performance to brute-force search with low computation complexity [155]. In a UAV-enabled MEC system, a deep RL framework based on ConvGNN is designed to online train continuous flight actions and task offloading [156]. In addition, GAT is exploited to learn cooperative link scheduling policies among users over a time-varying graph in a UAV-aided hybrid satellite-terrestrial network, aiming to alleviate masking effect in satellite links [157].

Lessons learned 6: GNNs efficiently deal with not only standard radio resource management issues but also aerial coverage planning due to UAV mobility, such as flying trajectory, hovering position, and swarm connectivity. These issues widely exist various UAV networks and are addressed by different GNN-based methods. ConvGNN exhibits strong universality and is capable of obtaining extra spatial information of UAV networks. It is shown that ConvGNN can perform comparably to brute-force search in a UAV-relaying network. GNN-based RL is utilized to improve cooperation among UAVs on trajectory optimization and access control, catering to UAV mobility and dynamic network topology. STGNN is used to capture spatial and temporal information of aerial cellular networks for optimizing UAV-BS coverage and user association.

F. Summary and Discussion

This section reviews the application of graph learning for resource management in wireless networks. We classify all the literature into five categories that power control, spectrum management, beamforming design, task scheduling, and aerial coverage planning. We first focus on how GNN motivates the development of power control in wireless communications. Interference-limited networks and RANs are main scenarios of existing GNN-based power control methods where ConvGNN and GNN-based unfolding are the most prevalent algorithmic frameworks. Next, we investigate graph learning methods for spectrum management issues including access control, channel allocation, and user association. ConvGNN, graph embedding methods, and GNN-based RL are primary approaches for designing effective spectrum management algorithms. Then, we elaborate on beamforming design using GNNs in MIMO interference channel, MU-MIMO systems, and RIS-aided MIMO systems. Furthermore, we review GNN-based task scheduling in advanced wireless technologies from the perspectives of content caching, computation offloading, and edge learning. ConvGNN, STGNN, and GNN-based RL are popular models and frameworks for task scheduling to meet various service requirements. Finally, we concentrate on the application of GNNs to aerial coverage planning. GNNs show the efficiency and practicality in aerial access control, UAV trajectory planning and deployment. In the future, the densification and heterogeneity of wireless networks will become increasingly pronounced, along with an abundant variety of use cases [158]–[160]. This shift will propel graph learning techniques, particularly GNNs, to become a crucial method for resource management, leveraging their advantages of scalability, efficiency, generalizability, and compatibility [161].

According to the literature review in this two-part survey, we have several remarks as follows.

- Graph optimization is suitable for resource management in moderate-scale and small-scale wireless networks to

Table IV
A SUMMARY OF APPROACHES USING GNN ARCHITECTURES FOR TASK SCHEDULING

Issues	References	Models	Categories	Learning Manners	Scopes
Content Caching	[125]	GAT based MARL	GNN-based RL	Soft actor-critic RL	Content placement and delivery
	[126]	GCN with LSTM	STGNN	Supervised	Content placement
Computation Offloading	[127], [128]	GCN	ConvGNN	Policy-based RL	Interdependent task
	[129]	GAT and LSTM based PPO	STGNN	Meta-learning	
	[130], [131]	GCN based DQN	GNN-based RL	Value-based RL	
	[132]	GCN	ConvGNN	Unsupervised	
	[133]	GraphSAGE and LSTM based DDPG	STGNN	Actor-critic RL	Independent task
	[134]	HetGNN based DQN	GNN-based RL	Value-based RL	
	[135]	GCN based DRL	GNN-based RL	Actor-critic RL	
	[136]	GAT based DQN	GNN-based RL	Value-based RL	
	[137]	GAT	ConvGNN	Unsupervised	
	[138]	MPNN with GRU	STGNN	Value-based RL	
	[139]	GAT based DRQN	GNN-based RL	Value-based RL	
	[140]	GCN for MARL	GNN-based RL	Actor-critic RL	
	[141]	MPNN	ConvGNN	Supervised	Resource slicing
ISAC	[142]	GAT for MARL	GNN-based RL	Actor-critic RL	Message routing
	[143]	MPNN	ConvGNN	Supervised	Link scheduling
	[144]	Dynamic GNN	ConvGNN	Supervised	
Edge Learning	[145]	GCN	ConvGNN	Unsupervised	Distributed DL
	[146]	GCN	ConvGNN	Primal-dual unsupervised	FL
	[147]	MPNN	ConvGNN	Unsupervised	
	[148]	GCN with LSTM	STGNN	Supervised	
	[149]	GCN based DQN	GNN-based RL	Value-based RL	

find the optimal or semi-optimal solution. The algorithm performance is sensitive to the accuracy of network state and channel condition.

- Graph learning is suitable for resource management in large-scale wireless networks to find the near-optimal or semi-optimal solution. GNN models have good scalability to network size and well generalization to network state.

IV. TECHNICAL CHALLENGES AND FUTURE DIRECTIONS

Diverse approaches reviewed in this two-part survey evidently show that graph optimization and learning can effectively address various emerging issues at the field of resource management in wireless networks. There are still existing challenges and new research directions which are discussed as follows.

A. Technical Challenges

1) Network Scale and Density: Wireless networks are growing in size and density, which contains a large number of wireless equipments. To effectively manage entire wireless networks, it is necessary to construct a large-scale graph representing the relationships among numerous equipments. The handling of such a large-scale graph requires extensive computation and storage, which is an open issue in the fields of graph theory and computer science. Furthermore, because the size of graph directly affects the computation complexity of algorithms, large-scale and dense wireless networks raise a challenge to the feasibility of graph optimization and learning approaches.

2) Dynamicity in Wireless Networks: Wireless networks are dynamic systems inherently. First, the condition of wireless channel varies randomly to cause fluctuations in the received signal power. This leads to the dynamicity of instantaneous network performance, such as throughput, delay, etc. Second, the mobility of wireless devices can change the network topology, leading to dynamic associations and interference among these devices. Furthermore, user requirements, traffics, and buffer states are all dynamic in practice. Although GNNs and distributed graph optimization algorithms demonstrate their adaptability to these dynamics, they still present challenges in the terms of model efficiency and generalization.

3) Heterogeneity of Wireless Devices: Recent wireless networks mainly consists of three types of wireless devices including server devices (e.g., edge servers), user devices (e.g., mobile terminals), and connecting devices (e.g., BSs and relays). In many resource management problems, these heterogeneous devices cannot be uniformly represented and differentiated by identical vertices and edges. Heterogeneous graph are increasingly utilized to depict wireless networks. However, most existing GNNs are primarily designed for homogeneous graphs and cannot directly applied over heterogeneous graphs. Even when adapted for heterogeneous graphs using additional operations, their performance is often limited. Therefore, developing GNNs over heterogeneous graphs for resource management remains a challenge.

4) Incompleteness of Network Data: Many resource management factors significantly affect wireless network performance, as reflected by several key parameter indicators (KPIs). It is labor-intensive and impractical to establish the relationships between each factor and KPIs, and label them for building training, validation, and test data sets. Incomplete data is typically used for unsupervised learning and reinforcement learning, which usually have a performance gap compared to supervised learning. Consequently, designing an efficient graph optimization and learning algorithm using incomplete

data poses a significant challenge in resource management.

B. Future Directions

1) Advanced Graphs for Wireless Networking: The advancement of graph theory introduces various new methods to model and optimize problems on resource management in wireless networks. For examples, extremal graph theory investigates how a parameter of a graph can be big or small under given constraints, such as the number of vertices and edges. It has potential benefits in proposing scalable graph optimization and learning approaches for resource management [162]. Random graph theory focuses on a graph obtained by randomly adding or deleting edges and investigates its probability distribution and stochastic properties at different stages. It provides an alternative way to depict the dynamicity of wireless networks [163]. In addition, the theory of games on graphs can be utilized to design distributed algorithms for resource management [164]. In consequence, advanced graph modeling and optimization approaches for wireless communications are worthy of investigation.

2) Scalable Resource Management with Graphs: Densification and heterogeneity are two significant features of future wireless networks to enhance network capacity and coverage performance. As network density and complexity increase, scalable resource management schemes become vital. Graph optimization and learning have shown benefits in scalability. However, striking a balance between scalability and effectiveness is challenging. For instance, the scalability of GNNs comes at the expense of graph completeness, such as the sampling operation in GraphSAGE that leads to partial loss of graph information. Therefore, improving the effectiveness of scalable wireless management with graphs is a promising direction.

3) Generative Models on Graphs for Wireless Networking: Scalability is one of primary reasons why GNNs become popular for resource management in wireless networks. However, this is just one aspect of their generalization and generative capability. Beyond network scale, the model output after changes in user requirements, fronthaul/backhaul capacity limitations, delay constraints and others is an important evaluation criterion for the model's performance. In order to enhance generalization and generative capability, GAEs, graph transformers, and other generative models are increasingly employed for resource management in wireless networks [165]–[169]. Therefore, the application of generative models on graph for wireless networking is a promising research direction.

4) Graph Learning with Domain Knowledge of Resource Management: Knowledge-driven machine learning is a growing field that improves learning efficiency and explainability by incorporating domain knowledge into model architecture and algorithm design. For resource management in wireless communications, domain knowledge includes information and experience of specific resource management problems, such as the water-filling algorithm for power allocation and semidefinite relaxation method for beamforming design. This knowledge can be embedded in models by customizing the input data, loss function, and model construction. Therefore, the incorporation of abundant resource management knowledge into graph learning is an important research direction.

V. Conclusion

This paper is the Part II of a two-part survey to present a comprehensive survey of graph learning for resource management in wireless networks, while the Part I focuses on graph optimization in this field. In this part, we have introduced the fundamentals of graph learning and presented several modern GNN models. Then, the state-of-the-art literature on graph learning methods for resource management has been comprehensively reviewed according to different issues, i.e., power control, spectrum management, beamforming design, task scheduling, and aerial coverage planning. Finally, we have discussed current technical challenges and future research directions of graph optimization and learning for resource management in future wireless networks.

References

[1] M. Eisen and A. Ribeiro, "Large Scale Wireless Power Allocation with Graph Neural Networks," in *IEEE SPAWC*, Jul. 2019, pp. 1–5.

[2] ——, "Optimal Wireless Resource Allocation With Random Edge Graph Neural Networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, Apr. 2020.

[3] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *IEEE GC Wkshps*, Dec. 2019, pp. 1–6.

[4] ——, "Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.

[5] Y. Shen, J. Zhang, S. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3554–3569, May 2023.

[6] Y. Shen, J. Zhang, and K. B. Letaief, "How neural architectures affect deep learning for communicationnetworks?" in *Proc. IEEE ICC*, May 2022, pp. 389–394.

[7] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, "Link Scheduling using Graph Neural Networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3997–4012, Jun. 2023.

[8] M. Lee, G. Yu, and G. Y. Li, "Wireless Link Scheduling for D2D Communications with Graph Embedding Technique," in *Proc. IEEE ICC*, Jun. 2020, pp. 1–6.

[9] ——, "Graph Embedding-Based Wireless Link Scheduling With Few Training Samples," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2282–2294, Apr. 2021.

[10] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sept. 2018.

[11] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Proc. NeurIPS*, vol. 30, Dec. 2017, pp. 3394–3404.

[12] P. Almasan, J. Suarez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, "Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," *Computer Communications*, vol. 196, pp. 184–194, Dec. 2022.

[13] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[14] C. Gallicchio and A. Micheli, "Graph Echo State Networks," in *Proc. IEEE IJCNN*, Jul. 2010, pp. 1–8.

[15] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning Steady-States of Iterative Algorithms over Graphs," in *Proc. PMLR ICML*, vol. 80, Jul. 2018, pp. 1106–1114.

[16] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated Graph Sequence Neural Networks," Sept. 2017, arXiv:1511.05493 [cs, stat].

[17] V. Zayats and M. Ostendorf, "Conversation Modeling on Reddit Using a Graph-Structured LSTM," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 121–132, Feb. 2018.

[18] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering," in *Proc. NeurIPS*, vol. 29, Dec. 2016, pp. 3844–3852.

[19] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive Graph Convolutional Neural Networks," *Proc. AAAI*, vol. 32, no. 1, Apr. 2018.

[20] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph Wavelet Neural Network," in *Proc. ICLR*, May 2019.

[21] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," in *Proc. PMLR ICML*, vol. 70. PMLR, Jul. 2017, pp. 1263–1272.

[22] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Proc. NeurIPS*, vol. 30, Dec. 2017, pp. 1025–1035.

[23] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," in *Proc. ICLR*, Apr. 2018.

[24] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" in *Proc. ICLR*, May 2019.

[25] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured Sequence Modeling with Graph Convolutional Recurrent Networks," in *Proc. NeurIPS*, Cham, Dec. 2018, pp. 362–373.

[26] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic Flow Prediction via Spatial Temporal Graph Neural Network," in *Proc. WWW*, Taipei Taiwan, Apr. 2020, pp. 1082–1092.

[27] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep Learning on Spatio-Temporal Graphs," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 5308–5317.

[28] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in *Proc. IJCAI*, Jul. 2018, pp. 3634–3640.

[29] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," *Proc. AAAI*, vol. 32, no. 1, Apr. 2018.

[30] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," *Proc. IJCAI*, pp. 1907–1913, May 2019.

[31] T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders," in *Bayesian Deep Learning Wkshps (NIPS)*, Nov. 2016.

[32] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially Regularized Graph Autoencoder for Graph Embedding," in *Proc. IJCAI*, Stockholm, Sweden, Jul. 2018, pp. 2609–2615.

[33] F. Faez, Y. Ommi, M. S. Baghshah, and H. R. Rabiee, "Deep Graph Generators: A Survey," *IEEE Access*, vol. 9, pp. 106 675–106 702, 2021.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proc. NeurIPS*, vol. 30, Dec. 2017, pp. 6000–6010.

[35] Z. Wu, P. Jain, M. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, "Representing Long-Range Context for Graph Neural Networks with Global Attention," in *Proc. NeurIPS*, vol. 34, Dec. 2021, pp. 13 266–13 279.

[36] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *Proc. ICCV*, Oct. 2021, pp. 12 939–12 948.

[37] J. Zhang, H. Zhang, C. Xia, and L. Sun, "Graph-Bert: Only Attention is Needed for Learning Graph Representations," Jan. 2020, arXiv:2001.05140 [cs, stat].

[38] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," in *Wkshps DLG-AAAI 2021*, Feb. 2021.

[39] M. S. Hussain, M. J. Zaki, and D. Subramanian, "Global self-attention as a replacement for graph convolution," in *Proc. ACM SIGKDD*, Aug. 2022, pp. 655–665.

[40] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do Transformers Really Perform Badly for Graph Representation?" in *Proc. NeurIPS*, vol. 34, Dec. 2021, pp. 28 877–28 888.

[41] J. Suarez-Varela, P. Almasan, M. Ferriol-Galmes, K. Rusek, F. Geyer, X. Cheng, X. Shi, S. Xiao, F. Scarselli, A. Cabellos-Aparicio, and P. Barlet-Ros, "Graph neural networks for communication networks: Context, use cases and opportunities," *IEEE Netw.*, vol. 37, no. 3, pp. 146–153, May 2023.

[42] M. Lee, G. Yu, H. Dai, and G. Y. Li, "Graph Neural Networks Meet Wireless Communications: Motivation, Applications, and Future Directions," *IEEE Wireless Commun. Lett.*, vol. 29, no. 5, pp. 12–19, Oct. 2022.

[43] Y. Yang, Z. Zhang, Y. Tian, R. Jin, and C. Huang, "Implementing Graph Neural Networks Over Wireless Networks via Over-the-Air Computing: A Joint Communication and Computation Framework," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 62–69, Jun. 2023.

[44] Y. He, X. Zhong, Y. Gan, H. Cui, and M. Guizani, "A DDPG Hybrid of Graph Attention Network and Action Branching for Multi-Scale End-Edge-Cloud Vehicular Orchestrated Task Offloading," *IEEE Wireless Commun.*, vol. 30, no. 4, pp. 147–153, Aug. 2023.

[45] N. Naderializadeh, M. Eisen, and A. Ribeiro, "Wireless Power Control via Counterfactual Optimization of Graph Neural Networks," in *IEEE SPAWC*, May 2020, pp. 1–5.

[46] Y. B. Uslu, N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "A state-augmented approach for learning optimal resource management decisions in wireless networks," *arXiv preprint*, Nov. 2022.

[47] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "State-augmented learnable algorithms for resource management in wireless networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 5898–5912, 2022.

[48] L. Chen, J. Zhu, and J. Evans, "Accelerating Graph Neural Networks via Edge Pruning for Power Allocation in Wireless Networks," *arXiv preprint*, May 2023.

[49] T. Perera, S. Atapattu, Y. Fang, P. Dharmawansa, and J. Evans, "FlexNet: A Graph Neural Network Approach to Resource Management in Flexible Duplex Networks," in *Proc. IEEE WCNC*, Mar. 2023, pp. 1–6.

[50] L. Chen, J. Zhu, and J. Evans, "Graph Neural Networks for Power Allocation in Wireless Networks with Full Duplex Nodes," *arXiv preprint*, Mar. 2023.

[51] X. Gao, J. Wang, and M. Zhou, "The Research of Resource Allocation Method Based on GCN-LSTM in 5G Network," *IEEE Commun. Lett.*, vol. 27, no. 3, pp. 926–930, Mar. 2023.

[52] Z. Wang, M. Eisen, and A. Ribeiro, "Decentralized Wireless Resource Allocation with Graph Neural Networks," in *Proc. IEEE ACSSC*, Pacific Grove, CA, USA, Nov. 2020, pp. 299–303.

[53] ——, "Unsupervised learning for asynchronous resource allocation in ad-hoc wireless networks," in *Proc. IEEE ICASSP*, Jun. 2021, pp. 8143–8147.

[54] ——, "Learning Decentralized Wireless Resource Allocations With Graph Neural Networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 1850–1863, Mar. 2022.

[55] I. Nikoloska and O. Simeone, "Fast power control adaptation via meta-learning for random edge graph neural networks," in *IEEE SPAWC*, Sept. 2021, pp. 146–150.

[56] ——, "Modular Meta-Learning for Power Control via Random Edge Graph Neural Networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 457–470, Aug. 2022.

[57] Y. Gu, C. She, Z. Quan, C. Qiu, and X. Xu, "Graph Neural Networks for Distributed Power Allocation in Wireless Networks: Aggregation Over-the-Air," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7551 – 7564, Nov. 2023.

[58] X. Zhang, H. Zhao, J. Xiong, X. Liu, H. Yin, X. Zhou, and J. Wei, "Decentralized routing and radio resource allocation in wireless ad hoc networks via graph reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 3, pp. 1146–1159, Jan. 2024.

[59] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[60] A. Chowdhury, G. Verma, C. Rao, A. Swami, and S. Segarra, "Unfolding WMMSE using graph neural networks for efficient power allocation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6004–6017, Sept. 2021.

[61] A. Chowdhury, F. Gama, and S. Segarra, "Stability Analysis of Unfolded WMMSE for Power Allocation," in *Proc. IEEE ICASSP*, May 2022, pp. 5298–5302.

[62] H. Yang, N. Cheng, R. Sun, W. Quan, R. Chai, K. Aldubaikhy, A. Alqasir, and X. Shen, "Knowledge-driven resource allocation for wireless networks: A WMMSE unrolled graph neural network approach," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 18 902–18 916, Feb. 2024.

[63] J. Guo and C. Yang, "Learning Power Control for Cellular Systems with Heterogeneous Graph Neural Network," in *Proc. IEEE WCNC*, Mar. 2021, pp. 1–6.

[64] ——, "Learning power allocation for multi-cell-multi-user systems with heterogeneous graph neural networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 884–897, Feb. 2022.

[65] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Adaptive Wireless Power Allocation with Graph Neural Networks," in *Proc. IEEE ICASSP*, May 2022, pp. 5213–5217.

[66] D. Abode, R. Adeogun, and G. Berardinelli, "Power Control for 6G Industrial Wireless Subnetworks: A Graph Neural Network Approach," in *Proc. IEEE WCNC*, Mar. 2023, pp. 1–6.

[67] L. Salaun, H. Yang, S. Mishra, and C. S. Chen, "A GNN Approach for Cell-Free Massive MIMO," in *Proc. IEEE GLOBECOM*, Dec. 2022, pp. 3053–3058.

[68] B. Li, G. Verma, and S. Segarra, "Graph-based Algorithm Unfolding for Energy-aware Power Allocation in Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1359–1373, Feb. 2023.

[69] B. Li, G. Verma, C. Rao, and S. Segarra, "Energy-Efficient Power Allocation in Wireless Networks using Graph Neural Networks," in *Proc. IEEE ACSSC*, Oct. 2021, pp. 732–736.

[70] V. Lima, M. Eisen, K. Gatsis, and A. Ribeiro, "Large-Scale Graph Reinforcement Learning in Wireless Control Systems," *arXiv preprint*, Apr. 2022.

[71] M. Eisen and A. Ribeiro, "Transferable Policies for Large Scale Wireless Networks with Graph Neural Networks," in *Proc. IEEE ICASSP*, May 2020, pp. 5040–5044.

[72] S. Jamshidiha, V. Pourahmadi, A. Mohammadi, and M. Bennis, "Link Activation Using Variational Graph Autoencoders," *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2358–2361, Jul. 2021.

[73] T. Chen, X. Zhang, M. You, G. Zheng, and S. Lambotharan, "A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1712–1724, Feb. 2022.

[74] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Learning Resilient Radio Resource Management Policies With Graph Neural Networks," *IEEE Trans. Signal Process.*, vol. 71, pp. 995–1009, Mar. 2023.

[75] Z. Wang, M. Bennis, and Y. Zhou, "Graph attention-based MADRL for access control and resource allocation in wireless networked control systems," *IEEE Trans. Wireless Commun.*, Aug. 2024, early access.

[76] R. Sambamoorthy, J. Mandalapu, S. S. Peruru, B. Jain, and E. Altman, "Graph Neural Network Based Scheduling: Improved Throughput Under a Generalized Interference Model," in *Proc. Springer VALUETOOLS*, vol. 404, Dec. 2021, pp. 144–153.

[77] Z. Zhao, A. Swami, and S. Segarra, "Distributed Link Sparsification for Scalable Scheduling Using Graph Neural Networks," in *Proc. IEEE ICASSP*, May 2022, pp. 5308–5312.

[78] Z. Zhao, G. Verma, A. Swami, and S. Segarra, "Delay-Oriented Distributed Scheduling Using Graph Neural Networks," in *Proc. IEEE ICASSP*, May 2022, pp. 8902–8906.

[79] S. Zhang, B. Yin, W. Zhang, and Y. Cheng, "Topology Aware Deep Learning for Wireless Network Optimization," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9791–9805, Nov. 2022.

[80] M. Yang, D. Zhai, R. Zhang, B. Li, L. Cai, and F. R. Yu, "Joint admission and power control for massive connections via graph neural network," *IEEE Trans. Veh. Technol.*, Feb. 2024, early access.

[81] S. He, S. Xiong, W. Zhang, Y. Yang, J. Ren, and Y. Huang, "GBLinks: GNN-Based Beam Selection and Link Activation for Ultra-Dense D2D mmWave Networks," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3451–3466, May 2022.

[82] M. Li, C. Chen, H. Wu, X. Guan, and X. Shen, "Edge-Assisted Spectrum Sharing for Freshness-Aware Industrial Wireless Networks: A Learning-Based Approach," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7737–7752, Sept. 2022.

[83] W. Jiajun, S. Chengjian, and Y. Chenyang, "On the size generalizibility of graph neural networks for learning resource allocation," *Sci. China Inf. Sci.*, vol. 67, no. 4, pp. 142 301:1–142 301:16, Mar. 2024.

[84] Y. Zhang, S. Yuan, L. Hu, W. Qie, and D. Guo, "Slice Resource Allocation Technology of Cognitive Wireless Network Based on NOMA," in *Proc. IEEE ICECE*, Dec. 2021, pp. 169–173.

[85] X. Du, T. Wang, Q. Feng, C. Ye, T. Tao, L. Wang, Y. Shi, and M. Chen, "Multi-agent reinforcement learning for dynamic resource management in 6G in-X subnetworks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1900–1914, Mar. 2023.

[86] Z. Gao, Y. Shao, D. Gunduz, and A. Prorok, "Decentralized Channel Management in WLANs with Graph Neural Networks," *arXiv preprint*, Oct. 2022.

[87] C. Wang, J. Peng, L. Cai, W. Liu, S. Li, H. He, and Z. Huang, "Proactive bandwidth allocation for V2X networks with multi-attentional deep graph learning," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8542–8555, Aug. 2024.

[88] K. Hou, Q. Xu, X. Zhang, Y. Huang, and L. Yang, "User Association and Power Allocation Based on Unsupervised Graph Model in Ultra-Dense Network," in *Proc. IEEE WCNC*, Mar. 2021, pp. 1–6.

[89] M. Randall, P. Belzarena, F. Larroca, and P. Casas, "Deep Reinforcement Learning and Graph Neural Networks for Efficient Resource Allocation in 5G Networks," in *Proc. IEEE LATINCOM*, Nov. 2022, pp. 1–6.

[90] W. Deng, Y. Liu, M. Li, and M. Lei, "GNN-Aided User Association and Beam Selection for mmWave-Integrated Heterogeneous Networks," *IEEE Wireless Commun. Lett.*, vol. 12, no. 11, pp. 1836–1840, Nov. 2023.

[91] H. Zhang, X. Ma, X. Liu, L. Li, and K. Sun, "GNN-Based Power Allocation and User Association in Digital Twin Network for the Terahertz Band," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3111–3121, Oct. 2023.

[92] V. Ranasinghe, N. Rajatheva, and M. Latva-aho, "Graph Neural Network Based Access Point Selection for Cell-Free Massive MIMO Systems," in *Proc. IEEE GLOBECOM*, Dec. 2021, pp. 01–06.

[93] X. Liu, C. She, Y. Li, and B. Vucetic, "Edge-Wise Gated Graph Neural Network for User Association in Massive URLLC," in *IEEE GC Wkshps*, Dec. 2021, pp. 1–6.

[94] Z. Zhang, Q.-S. Hua, X. Zhang, H. Jin, and X. Liao, "DAG Scheduling with Communication Delays Based on Graph Convolutional Neural Network," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–16, Jul. 2022.

[95] X. Cai, J. Sheng, Y. Wang, B. Ai, and C. Wu, "A Novel Opportunistic Access Algorithm Based on GCN Network in Internet of Mobile Things," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 631–11 642, Jul. 2023.

[96] Z. Mei, X. Cai, J. Sheng, and C. Wu, "Heterogeneous Network Desion Algorithm Based on GCN Features Extraction in Railway Communication Scenarios," in *Proc. ComComAp*, Nov. 2021, pp. 71–76.

[97] X. Zhang, H. Zhao, J. Xiong, X. Liu, L. Zhou, and J. Wei, "Scalable Power Control/Beamforming in Heterogeneous Wireless Networks with Graph Neural Networks," in *Proc. IEEE GLOBECOM*, Dec. 2021, pp. 01–06.

[98] Y. Wang, Y. Li, Q. Shi, and Y.-C. Wu, "ENGNN: A General Edge-Update Empowered GNN Architecture for Radio Resource Management in Wireless Networks," *IEEE Trans. Wireless Commun.*, Oct. 2023, early access.

[99] Q. Hou, M. Lee, G. Yu, and Y. Cai, "Learning to Optimize Resource in Dynamic Wireless Environment via Meta-Gating Graph Neural Network," in *Proc. ISWCS*, Oct. 2022, pp. 1–6.

[100] T.-Y. Wang, H. Zhou, R. Kannan, A. Swami, and V. Prasanna, "Throughput optimization in heterogeneous MIMO networks: a GNN-based approach," in *ACM CoNEXT Wkshps*, Rome Italy, Dec. 2022, pp. 42–47.

[101] A. Chowdhury, G. Verma, A. Swami, and S. Segarra, "Deep Graph Unfolding for Beamforming in MU-MIMO Interference Networks," *arXiv preprint*, Apr. 2023.

[102] Y. Gu, C. She, S. Bi, Z. Quan, and B. Vucetic, "Graph neural network for distributed beamforming and power control in massive URLLC networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9099–9112, Feb. 2024.

[103] C. He, Y. Li, Y. Lu, B. Ai, Z. Ding, and D. Niyato, "ICNet: GNN-enabled beamforming for MISO interference channels with statistical CSI," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 12 225–12 230, Mar. 2024.

[104] J. Kim, H. Lee, S.-E. Hong, and S.-H. Park, "A Bipartite Graph Neural Network Approach for Scalable Beamforming Optimization," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 333–347, Jan. 2023.

[105] S. He, S. Xiong, Z. An, W. Zhang, Y. Huang, and Y. Zhang, "An Unsupervised Deep Unrolling Framework for Constrained Optimization Problems in Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8552–8564, Oct. 2022.

[106] Y. Li, Y. Lu, R. Zhang, B. Ai, and Z. Zhong, "Deep Learning for Energy Efficient Beamforming in MU-MISO Networks: A GAT-Based Approach," *IEEE Wireless Commun. Lett.*, vol. 12, no. 7, pp. 1264–1268, Jul. 2023.

[107] Y. Li, Y. Lu, B. Ai, O. A. Dobre, Z. Ding, and D. Niyato, "GNN-based beamforming for sum-rate maximization in MU-MISO networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9251–9264, Feb. 2024.

[108] Y. Li, Y. Lu, B. Ai, Z. Zhong, D. Niyato, and Z. Ding, "GNN-enabled max-min fair beamforming," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 12 184–12 188, Mar. 2024.

[109] Z. Song, Y. Lu, X. Chen, B. Ai, Z. Zhong, and D. Niyato, "A deep learning framework for physical-layer secure beamforming," *IEEE Trans. Veh. Technol.*, Aug. 2024, early access.

[110] S. He, J. Yuan, Z. An, W. Huang, Y. Huang, and Y. Zhang, "Joint User Scheduling and Beamforming Design for Multiuser MISO Downlink Systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 2975–2988, May 2023.

[111] S. Shrestha, X. Fu, and M. Hong, "Optimal Solutions for Joint Beamforming and Antenna Selection: From Branch and Bound to Graph Neural Imitation Learning," *IEEE Trans. Signal Process.*, vol. 71, pp. 831–846, Feb. 2023.

[112] ——, "Towards Efficient and Optimal Joint Beamforming and Antenna Selection: A Machine Learning Approach," in *Proc. IEEE ICASSP*, Jun. 2023, pp. 1–5.

[113] Z. Zhang, M. Tao, and Y.-F. Liu, "Learning to beamform in joint multicast and unicast transmission with imperfect csi," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2711–2723, Mar. 2023.

[114] L. Schynol and M. Pesavento, "Coordinated Sum-Rate Maximization in Multicell MU-MIMO With Deep Unrolling," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1120–1134, Apr. 2023.

[115] X. Xu, Y. Liu, Q. Chen, X. Mu, and Z. Ding, "Distributed Auto-Learning GNN for Multi-Cell Cluster-Free NOMA Communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1243–1258, Apr. 2023.

[116] T. Jiang, H. V. Cheng, and W. Yu, "Learning to Reflect and to Beamform for Intelligent Reflecting Surface With Implicit Channel Estimation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1931–1945, Jul. 2021.

[117] Z. Zhang, T. Jiang, and W. Yu, "Learning Based User Scheduling in Reconfigurable Intelligent Surface Assisted Multiuser Downlink," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 1026–1039, Aug. 2022.

[118] ——, "User Scheduling Using Graph Neural Networks for Reconfigurable Intelligent Surface Assisted Multiuser Downlink Communications," in *Proc. IEEE ICASSP*, May 2022, pp. 8892–8896.

[119] B. Zhao and C. Yang, "Learning Beamforming for RIS-aided Systems with Permutation Equivariant Graph Neural Networks," in *Proc. IEEE VTC-Spring*, Jun. 2023, pp. 1–5.

[120] J. Yang, J. Xu, Y. Zhang, H. Zheng, and T. Zhang, "Graph attention network-based precoding for reconfigurable intelligent surfaces aided wireless communication systems," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 9098–9102, Jan. 2024.

[121] A. Mehrabian and V. W. S. Wong, "Joint spectrum, precoding, and phase shifts design for ris-aided multiuser mimo thz systems," *IEEE Trans. Commun.*, vol. 72, no. 8, pp. 5087–5101, Mar. 2024.

[122] X. Xu, Q. Chen, X. Mu, Y. Liu, and H. Jiang, "Graph-Embedded Multi-Agent Learning for Smart Reconfigurable THz MIMO-NOMA Networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 259–275, Jan. 2022.

[123] M. Liu, C. Huang, M. Di Renzo, M. Debbah, and C. Yuen, "Cooperative Beamforming and RISs Association for Multi-RISs Aided Multi-Users MmWave MIMO Systems through Graph Neural Networks," *arXiv preprint*, Feb. 2023.

[124] B.-J. Chen, R. Y. Chang, F.-T. Chien, and H. V. Poor, "Energy-efficient multi-RIS-aided rate-splitting multiple access: A graph neural network approach," *IEEE Wireless Commun. Lett.*, vol. 13, no. 7, pp. 2003–2007, May 2024.

[125] Y. Yan, B. Zhang, C. Li, and C. Su, "Cooperative caching and fetching in D2D communications - a fully decentralized multi-agent reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16 095–16 109, Dec. 2020.

[126] D. Li, H. Zhang, H. Ding, T. Li, D. Liang, and D. Yuan, "User-Preference-Learning-Based Proactive Edge Caching for D2D-Assisted Wireless Networks," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 922–11 937, Jul. 2023.

[127] T. Wu, W. Jing, X. Wen, Z. Lu, and S. Zhao, "A Scalable Computation Offloading Scheme for MEC Based on Graph Neural Networks," in *IEEE GC Wkshps*, Dec. 2021, pp. 1–6.

[128] S. Yang, L. Zhang, L. Cui, Q. Dong, W. Xiao, and C. Luo, "RLCS: Towards a robust and efficient mobile edge computing resource scheduling and task offloading system based on graph neural network," *Computer Communications*, vol. 206, pp. 38–50, Jun. 2023.

[129] Y. Li, J. Li, Z. Lv, H. Li, Y. Wang, and Z. Xu, "GASTO: A Fast Adaptive Graph Learning Framework for Edge Computing Empowered Task Offloading," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 932–944, Jun. 2023.

[130] C.-T. Mo, J.-H. Chen, and W. Liao, "Graph Convolutional Network Augmented Deep Reinforcement Learning for Dependent Task Offloading in Mobile Edge Computing," in *Proc. IEEE WCNC*, Mar. 2023, pp. 1–6.

[131] J. Li, F. Zhou, W. Li, M. Zhao, X. Yan, Y. Xi, and J. Wu, "Componentized Task Scheduling in Cloud-Edge Cooperative Scenarios Based on GNN-enhanced DRL," in *Proc. IEEE/IFIP NOMS*, May 2023, pp. 1–4.

[132] M. Ma, "Multi-Task Offloading via Graph Neural Networks in Heterogeneous Multi-access Edge Computing," *arXiv preprint*, Jun. 2023, arXiv:2306.10232 [cs, eess].

[133] Y. Deng, H. Jiang, P. Cai, T. Wu, P. Zhou, B. Li, H. Lu, J. Wu, X. Chen, and K. Wang, "Resource Provisioning for Mitigating Edge DDoS Attacks in MEC-Enabled SDVN," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24 264–24 280, Dec. 2022.

[134] T. Pamuklu, A. Syed, W. S. Kennedy, and M. Erol-Kantarci, "Heterogeneous GNN-RL Based Task Offloading for UAV-aided Smart Agriculture," *IEEE Netw. Lett.*, vol. 5, no. 4, pp. 213–217, Dec. 2023.

[135] Z. Sun, Y. Mo, and C. Yu, "Graph-Reinforcement-Learning-Based Task Offloading for Multiaccess Edge Computing," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3138–3150, Feb. 2023.

[136] D. Wang, Y. Bai, G. Huang, B. Song, and F. R. Yu, "Cache-Aided MEC for IoT: Resource Allocation Using Deep Graph Reinforcement Learning," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 486–11 496, Jul. 2023.

[137] X. Wang, N. Cheng, L. Fu, W. Quan, R. Sun, Y. Hui, T. Luan, and X. S. Shen, "Scalable resource management for dynamic MEC: An unsupervised link-output graph neural network approach," in *Proc. IEEE PIMRC*, Sept. 2023, pp. 1–6.

[138] G. Wang, P. Cheng, Z. Chen, B. Vucetic, and Y. Li, "Inverse Reinforcement Learning with Graph Neural Networks for Full-Dimensional Task Offloading in Edge Computing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 6490 – 6507, Oct. 2023.

[139] Z. Yao, S. Xia, Y. Li, and G. Wu, "Cooperative Task Offloading and Service Caching for Digital Twin Edge Networks: A Graph Attention Multi-Agent Reinforcement Learning Approach," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3401–3413, Nov. 2023.

[140] Y. Yang, L. Feng, Y. Sun, Y. Li, W. Li, and M. A. Imran, "Multi-cluster cooperative offloading for VR task: A MARL approach with graph embedding," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8773–8788, Jan. 2024.

[141] A. Asheralieva, D. Niyato, and Y. Miyanaga, "Efficient Dynamic Distributed Resource Slicing in 6G Multi-Access Edge Computing Networks with Online ADMM and Message Passing Graph Neural Networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2614–2638, Apr. 2024.

[142] J. Lee, Y. Cheng, D. Niyato, Y. L. Guan, and D. Gonzalez G., "Intelligent resource allocation in joint radar-communication with graph neural networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 11 120–11 135, Oct. 2022.

[143] X. Li, M. Chen, Y. Liu, Z. Zhang, D. Liu, and S. Mao, "Graph neural networks for joint communication and sensing optimization in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3893–3907, Oct. 2023.

[144] X. Li, M. Chen, Y. Hu, Z. Zhang, D. Liu, and S. Mao, "Jointly optimizing terahertz based sensing and communications in vehicular networks: A dynamic graph neural network approach," *IEEE Trans. Wireless Commun.*, May 2024, early access.

[145] W. Qu, X. Ding, K. Yang, Y. Bao, and W. Chen, "ICTA: Intelligent Computing Task Allocation for Efficient Deep Learning in Distributed Edge Computing System of IoT," in *Proc. ACM BDSIC*, Xiamen China, Dec. 2020, pp. 61–69.

[146] B. Li, A. Swami, and S. Segarra, "Power Allocation for Wireless Federated Learning Using Graph Neural Networks," in *Proc. IEEE ICASSP*, May 2022, pp. 5243–5247.

[147] Z. Wang, Y. Zhou, Y. Zou, Q. An, Y. Shi, and M. Bennis, "A Graph Neural Network Learning Approach to Optimize RIS-Assisted Federated Learning," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 6092–6106, Sept. 2023.

[148] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated Learning in the Sky: Aerial-Ground Air Quality Sensing Framework With UAV Swarms," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021.

[149] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an Automated Auction Framework for Wireless Federated Learning Services Market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.

[150] Z. Mou, F. Gao, J. Liu, and Q. Wu, "Resilient UAV Swarm Communications With Graph Convolutional Neural Network," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 393–411, Jan. 2022.

[151] X. Zhang, H. Zhao, J. Wei, C. Yan, J. Xiong, and X. Liu, "Cooperative Trajectory Design of Multiple UAV Base Stations With Heterogeneous Graph Neural Networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1495–1509, Mar. 2023.

[152] Z. Ye, K. Wang, Y. Chen, X. Jiang, and G. Song, "Multi-UAV Navigation for Partially Observable Communication Coverage by Graph Reinforcement Learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 4056–4069, Jul. 2023.

[153] Z. Feng, D. Wu, M. Huang, and C. Yuen, "Graph-attention-based reinforcement learning for trajectory design and resource assignment in multi-UAV-assisted communication," *IEEE Internet Things J.*, vol. 11, no. 16, pp. 27 421–27 434, May 2024.

[154] Y. Pan, X. Wang, Z. Xu, N. Cheng, W. Xu, and J.-j. Zhang, "GNN-empowered effective partial observation MARL method for AoI management in multi-UAV network," *IEEE Internet Things J.*, Aug. 2024, early access.

[155] X. Wang, L. Fu, N. Cheng, R. Sun, T. Luan, W. Quan, and K. Aldubaikhy, "Joint Flying Relay Location and Routing Optimization for 6G UAV-IoT Networks: A Graph Neural Network-Based Approach," *Remote Sensing*, vol. 14, no. 17, p. 4377, Jan. 2022.

[156] K. Li, W. Ni, X. Yuan, A. Noor, and A. Jamalipour, "Exploring Graph Neural Networks for Joint Cruise Control and Task Offloading in UAV-enabled Mobile Edge Computing," in *Proc. IEEE VTC-Spring*, Jun. 2023, pp. 1–6.

[157] Y.-J. Chen, W. Chen, and M.-L. Ku, "Trajectory Design and Link Selection in UAV-Assisted Hybrid Satellite-Terrestrial Network," *IEEE Commun. Lett.*, vol. 26, no. 7, pp. 1643–1647, Jul. 2022.

[158] B. Qian, H. Zhou, T. Ma, K. Yu, Q. Yu, and X. Shen, "Multi-operator spectrum sharing for massive IoT coexisting in 5G/B5G wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 881–895, Mar. 2021.

[159] J. He, N. Cheng, Z. Yin, H. Zhou, C. Zhou, K. Aldubaikhy, A. Alqasir, and X. Shen, "Load-aware network resource orchestration in LEO satellite network: A GAT-based approach," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15 969–15 984, Jan. 2024.

[160] A. Asheralieva, D. Niyato, and Y. Miyanaga, "Efficient dynamic distributed resource slicing in 6G multi-access edge computing networks with online ADMM and message passing graph neural networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2614–2638, Mar. 2024.

[161] Z. Liu, J. Zhang, E. Shi, Z. Liu, D. Niyato, B. Ai, and X. S. Shen, "Graph neural network meets multi-agent reinforcement learning: Fundamentals, applications, and future directions," *IEEE Wireless Commun.*, Aug. 2024, early access.

[162] Y.-C. Chan, J. Wang, and S. A. Jafar, "Toward an Extremal Network Theory-Robust GDoF Gain of Transmitter Cooperation Over TIN," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3827–3845, Jun. 2020.

[163] M. Sood and O. Yagan, "On the Minimum Node Degree and k-Connectivity in Inhomogeneous Random K-Out Graphs," *IEEE Trans. Inf. Theory*, vol. 67, no. 10, pp. 6868–6893, Oct. 2021.

[164] N. Fijalkow, N. Bertrand, P. Bouyer-Decitre, R. Brenguier, A. Carayol, J. Fearnley, H. Gimbert, F. Horn, R. Ibsen-Jensen, N. Markey, B. Monmege, P. Novotny, M. Randour, O. Sankur, S. Schmitz, O. Serre, and M. Skomra, "Games on graphs," May 2023, arXiv:2305.10546 [cs].

[165] M. Xu, H. Du, D. Niyato, J. Kang, Z. Xiong, S. Mao, Z. Han, A. Jamalipour, D. I. Kim, X. Shen, V. C. M. Leung, and H. V. Poor, "Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1127–1170, Jan. 2024.

[166] Y. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, and A. Jamalipour, "Deep generative model and its applications in efficient wireless network management: A tutorial and case study," *IEEE Wireless Commun.*, vol. 31, no. 4, pp. 199–207, Apr. 2024.

[167] H. Du, R. Zhang, Y. Liu, J. Wang, Y. Lin, Z. Li, D. Niyato, J. Kang, Z. Xiong, S. Cui, B. Ai, H. Zhou, and D. I. Kim, "Enhancing deep reinforcement learning: A tutorial on generative diffusion models in network optimization," *IEEE Commun. Surveys Tuts.*, May 2024, early access.

[168] C. Zhao, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, X. Shen, and K. B. Letaief, "Generative AI for secure physical layer communications: A survey," *IEEE Trans. Cogn. Commun. Netw*, Aug. 2024, early access.

[169] H. Du, D. Niyato, J. Kang, Z. Xiong, P. Zhang, S. Cui, X. Shen, S. Mao, Z. Han, A. Jamalipour, H. Vincent Poor, and D. I. Kim, "The age of generative AI and AI-generated everything," *IEEE Netw.*, Jul. 2024, early access.

Yanpeng Dai (S'19-M'20) received the B.Eng. degree from Shandong Normal University, Jinan, China, in 2014, and the Ph.D. degree from Xidian University, Xi'an, China, in 2020. He is currently an Associate Professor with the School of Information Science and Technology, Dalian Maritime University, Dalian, China. He was a Visiting Student at the University of Waterloo, Waterloo, ON, Canada. His research interests include resource management and interference coordination for heterogeneous wireless networks and maritime communication systems.

Ling Lyu (M'19) received the B.S. degree in telecommunication engineering from Jinlin University, Changchun, China, in 2013; and the Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019. She joined Dalian Maritime University, China, in 2019, where she is currently an associate professor with the School of Information Science and Technology. She was a visiting student in University of Waterloo, Canada (September 2017- September 2018). Her current research interests include wireless sensor and actuator network and application in industrial automation, the joint design of communication and control in industrial cyber-physical systems, estimation and control over lossy wireless networks, machine type communication enabled reliable transmission in the fifth generation network, resource allocation, energy efficiency.

Nan Cheng (M'16-SM'23) received the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo in 2016, and B.E. degree and the M.S. degree in Information and Telecommunications Engineering from Tongji University in 2009 and 2012, respectively. He worked as a Postdoctoral fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2019. He is currently a professor with State Key Lab. of ISN and with School of Telecommunications Engineering, Xidian University, Shaanxi, China. He has published over 90 journal papers in IEEE Transactions and other top journals. He serves as associate editors for IEEE Internet of Things Journal, IEEE Transactions on Vehicular Technology, IEEE Open Journal of the Communications Society, and Peer-to-Peer Networking and Applications, and serves/served as guest editors for several journals. His current research focuses on B5G/6G, AI-driven future networks, and space-air-ground integrated network.

Min Sheng (M'03-SM'16) received the M.S. and Ph.D. degrees in communication and information systems from Xidian University, Shaanxi, China, in 2000 and 2004, respectively. She is currently a Full Professor and Director with the State Key Laboratory of Integrated Service Networks, Xidian University. Her general research interests include mobile ad hoc networks, 5G mobile communication systems and satellite communications networks. She is Fellow of China Institute of Electronics (CIE) and Fellow of China Institute of Communications (CIC).

Junyu Liu (S'15-M'17) received his B.Eng. and Ph.D. degrees in communication and information systems from Xidian University, Shaanxi, China, in 2007 and 2016, respectively. He is currently a Full Professor with the State Key Laboratory of Integrated Service Networks, Xidian University. His research interests include wireless coverage and networking technology in heterogeneous networks.

Xiucheng Wang is currently pursuing a Ph.D degree from Xidian University. His research area of interest is machine learning of the wireless network.

Shuguang Cui (S'99-M'05-SM'12-F'14) received his Ph.D in Electrical Engineering from Stanford University, California, USA, in 2005. Afterwards, he has been working as assistant, associate, full, Chair Professor in Electrical and Computer Engineering at the University of Arizona, Texas A&M University, UC Davis, and CUHK at Shenzhen, respectively. He has also served as the Executive Dean for the School of Science and Engineering at CUHK, Shenzhen, the Executive Vice Director at Shenzhen Research Institute of Big Data, and the Director for Future Network of Intelligence Institute (FNii). His current research interests focus on the merging between AI and communication networks. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds' Most Influential Scientific Minds by ScienceWatch in 2014. He was the recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and TPC co-chairs for many IEEE conferences. He has also been serving as the area editor for IEEE Signal Processing Magazine, and associate editors for IEEE Transactions on Big Data, IEEE Transactions on Signal Processing, IEEE JSAC Series on Green Communications and Networking, and IEEE Transactions on Wireless Communications. He has been the elected member for IEEE Signal Processing Society SPCOM Technical Committee (2009~2014) and the elected Chair for IEEE ComSoc Wireless Technical Committee (2017~2018). He is a member of the Steering Committee for IEEE Transactions on Big Data and the Chair of the Steering Committee for IEEE Transactions on Cognitive Communications and Networking. He is also the Vice Chair of the IEEE VT Fellow Evaluation Committee and a member of the IEEE ComSoc Award Committee. He was elected as an IEEE Fellow in 2013, an IEEE ComSoc Distinguished Lecturer in 2014, and IEEE VT Society Distinguished Lecturer in 2019. In 2020, he won the IEEE ICC best paper award, ICIP best paper finalist, the IEEE Globecom best paper award. In 2021, he won the IEEE WCNC best paper award. In 2023, he won the IEEE Marconi Best Paper Award, got elected as a Fellow of both Canadian Academy of Engineering and the Royal Society of Canada, and started to serve as the Editor-in-Chief for IEEE Transactions on Mobile Computing.

Lin Cai (S'00-M'06-SM'10-F'20) has been with the Department of Electrical & Computer Engineering at the University of Victoria since 2005, and she is currently a Professor. She is an NSERC E.W.R. Steacie Memorial Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and an IEEE Fellow. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and the Internet of Things. She has been elected to serve the board of the IEEE Vehicular Technology Society, 2019 - 2024, and as its VP in Mobile Radio. She has been a Board Member of IEEE Women in Engineering (2022-24) and IEEE Communications Society (2024-2026). She has served as an Associate Editor-in-Chief for IEEE Transactions on Vehicular Technology, and as a Distinguished Lecturer of the IEEE VTS Society and the IEEE Communications Society.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and an International Fellow of the Engineering Academy of Japan.

Dr. Shen received "West Lake Friendship Award" from Zhejiang Province in 2023, President's Excellence in Research from the University of Waterloo in 2022, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He serves/served as the General Chair for the 6G Global Conference'23, and ACM Mobihoc'15, Technical Program Committee Chair/Co-Chair for IEEE Globecom'24, 16 and 07, IEEE Infocom'14, IEEE VTC'10 Fall, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. Dr. Shen is the Past President of the IEEE ComSoc, the Vice President for Technical & Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, and Member of IEEE Fellow Selection Committee of the ComSoc. Dr. Shen served as the Editor-in-Chief of the IEEE Internet of Things Journal, IEEE Network, and Peer-to-Peer Networking and Applications.