Access Point Selection and Beamforming Design for Cell-Free Network: From Fractional Programming to GNN

Xuanhong Yan, *Student Member, IEEE*, Zheng Wang, *Senior Member, IEEE* ,
Yi Jia, Zhengming Zhang, *Student Member, IEEE*,
Yongming Huang, *Senior Member, IEEE*

***Abstract*— In this paper, the cross-layer optimization problem of access point selection (APS) and beamforming (BF) in cell-free network (CFN) with local CSI is studied, where constraints of per AP power and the number of active APs are considered. Such a joint APS&BF optimization problem is modeled as a mixed-integer nonlinear programming (MINP) problem aiming at maximizing the sum rate of the whole system. Fractional programming (FP)-based and alternating optimization (AO)-based algorithms with weighted l_1-norm approximation are proposed to solve this MINP problem. However, the latter performs better than the former, with higher complexity. A lightweight multi-head single-body graph neural network (MHSB-GNN) algorithm is proposed, where the nodes and structures are innovatively designed. The MHSB-GNN benefits from the different node updating modules for different user equipment (UE), which introduce extra prior information into the graph and mine specific information of different UEs. Moreover, the equivalence between GNN and FP-based algorithm is proved to provide interpretability and theoretical guarantees for MHSB-GNN. The analysis of convergence and complexity validates the accuracy and effectiveness of the FP and AO-based algorithms. Leveraging the existing APS and BF solver, it is shown that the three proposed algorithms guarantee comparable performance as the exhaustive search algorithm in performance and complexity.**

***Index Terms*—User-centric cell free network, cross-layer optimization, AP selection, beamforming, MINP, GNN.**

I. Introduction

Communication networks never stop evolving, whether in the past, present, or future. The fifth generation (5G) can no longer meet the data transmission rate requirements of a fully digitalized world. As a result, academia and industry are motivated to start innovative research to shape the vision of future sixth generation (6G) networks that are well summarized in [1], [2]. Cell-free network (CFN) is one of the novel network technologies of 6G since this architecture offers high spectral efficiency (SE) and coverage probability required in 6G [3], [4]. However, with the existence of a large number of fronthaul links, significant interference, and high synchronization requirement, non-coherent user-centric CFN is applied to overcome these challenges by selecting the serving access points (APs) for each user equipment (UE) without sacrificing too much performance [5], [6].

To ensure the effective deployment of user-centric CFN, the design of AP selection (APS) in the control layer and beamforming (BF) in the physical layer is one of the challenging problems [7]. Jointly designing APS and BF brings better performance than separately designing APS and BF because the decoupled issues cannot ensure the optimality of the APS&BF problem [7]–[11]. However, the cross-layer optimization problem of APS&BF is a mixed-integer nonlinear programming (MINP) problem, which is NP-hard. Solutions for the MINP problem benefit from these studies of joint design of base station clustering and BF in communication systems [12]–[14]. Exhaustive search (ES) and Branch and bound (B&B), which are usually treated as baselines, can obtain the optimal solution of MINP problem [15]–[18]. However, the complexity of ES and B&B is exceptionally high because they traverse the feasible domain of every possible integer variable.

To further decrease the computational complexity of algorithms, researchers investigate suboptimal solutions to solve the MINP problem. A key point to solve the MINP problem is the transformation of the integer variables. Smoothed l_0-norm approximation [13] and reweighted l_1 transform [19] are effective transforms to make the MINP problem a common continuous optimization problem. Then the rest beamforming design can be solved by typical algorithms, e.g., weighted minimum mean squared error (WMMSE) [20], fractional programming (FP) algorithms [21] and convex-concave procedure (CCP) [13], [22]. However, these sub-optimal solutions trade the performance for ease of implementation. Besides, the direct form FP-based algorithm in [23], [24] adds a significant complexity burden by introducing an additional convex subproblem in each iteration, where such an introduced subproblem still needs to be solved iteratively.

Recently, machine learning [25] has demonstrated great potential in solving the communication problems, such as channel estimation [26], beamforming [27], [28], beam tracking [29] and so on. The primary research area is learning-to-optimize, which aims to exhibit promising results with affordable computational complexity [30]. The most straightforward

This work was supported by the National Natural Science Foundation of China under Grants 62225107 and 62371124, the Natural Science Foundation on Frontier Leading Technology Basic Research Project of Jiangsu under Grant BK20222001, the Fundamental Research Funds for the Central Universities under Grant 2242022k60002, and the Major Key Project of PCL. *(Corresponding author: Yongming Huang.)*

The authors are with the School of Information Science and Engineering and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, and also with the Pervasive Communications Center, Purple Mountain Laboratories, Nanjing 211111, China (e-mail: {yanxuanhong; zheng_wang; jiay; zmzhang; huangym}@seu.edu.cn).

way to implement learning-to-optimize is end-to-end learning, where the input-output mapping of the optimization problem is learned directly. Some classic architectures of DNN (i.e., fully connected networks (FCNs) and convolutional neural networks (CNNs)) achieve near-optimal performance by either supervised learning or unsupervised learning. For the joint APS&BF problem, the authors in [31], [32] proposed to train a DNN by unsupervised learning or supervised learning to achieve the near-optimal sum-rate in CFN. Note that the works [31], [32] treat the optimization problems as a black box and thus lack interpretability. Moreover, although these classic architectures achieve near-optimal performance and fast execution on small-scale wireless networks, the performance is not guaranteed when the number of samples for training is reduced. For example, the authors in [33] verify that the number of training samples for FCNs/CNNs is huge when such networks are able to achieve comparable performance.

Researchers model the wireless system as a graph to address the above issues and develop the graph neural network (GNN). Specifically, researchers in [34] proposed a GNN-based algorithm to design beamforming in an intelligent reflecting surface (IRS) system by learning the mapping from the received pilots to beamforming vectors and reflecting coefficients. In [33], the authors give the theoretical underpinnings and design guidelines when GNN is applied to communications and prove the equivalence between message passing GNN and iterative optimization algorithms. Researchers in [35] present an unsupervised GNN structure to finish the joint beam selection and link activation in ultra-dense device-to-device (D2D) mmWave communication networks. Different from the FCNs/CNNs, the adopted GNNs in [33]–[36] incorporate the prior domain knowledge (i.e., the network topology information) of wireless networks by modeling the UE/AP as the node and the interference relationship between UEs as the edge of the graph. This kind of modeling approach increases the interpretability of the NNs. Another remarkable property of GNN is permutation invariant that the outputs of GNN are regardless of the indices of the inputs. Moreover, [33] proves the advantages in the scalability and sample efficiency of GNNs over FCNs/CNNs theoretically. One main issue of the above GNN-based approaches is that they are not applied to large-scale systems because the increased network size makes it harder to train GNNs.

Distributed implementation of GNNs helps GNNs be applied to multi-cell systems and reduces the interaction overhead among cells. In [37], researchers propose a distributed GNN-based approach for IRS-enhanced cell-free MIMO networks by learning the mapping from the local channels to beamformers and reflecting coefficients. Researchers in [38] propose a novel communication-efficient distributed Auto-GNN architecture to reduce computation and information exchange burdens. Researchers in [32] propose a novel scheme, distributed learning for uplink cell-free massive MIMO beamforming, which can achieve multi-AP cooperation without explicitly estimating their CSI. Since this distributed implementation of the network architecture requires local channel information and statistical interference channel information as inputs, this ensures the application of GNNs in large-scale systems. However, these applications are essentially one-AP-to-one-UE service relationships, and there is no way to solve the problem of multi-AP-to-multi-UE beamforming design in non-coherent transmission CFNs.

This paper aims to jointly design APS&BF with the power constraint and the number of received data streams per UE. Motivated by the advantages of the distributed GNN-based algorithm over the centralized GNN-based algorithm, we propose APS&BF schemes with local CSI. This paper proposes two optimization-based approaches and one learning-based approach to design APS&BF with local CSI jointly. We adopt non-coherent transmission to ensure synchronization as in our published conference paper [39]. Each UE adopts successive interference cancellation (SIC) to decode the singles. As in [37], [38], the joint APS&BF design in this work is based on local CSI, which brings lower interaction overhead and practicality. Different from [34], [37], the GNN-based beamforming design scheme studied in this paper is suitable for multi-cell multi-AP multi-UE scenarios. To summarize, the major contributions of this paper are outlined as follows:

1) Three algorithms are proposed to design APS&BF with local CSI, which brings lower interaction overhead. FP-based and alternating optimization (AO)-based algorithms are derived based on the weighted l_1-norm approximation, Lagrange dual transform, and quadratic transform.

2) This paper also proposes a novel multi-head single-body structure GNN-based (MHSB-GNN) algorithm and proves the equivalence between GNN and FP-based algorithms, providing interpretability and theoretical guarantees for MHSB-GNN. In this work, we innovatively model the beams as nodes of the graph and treat the neighboring group as another kind of node, providing the inter-group interfering information. The critical design of MHSB-GNN lies in the fact that the node updating modules belonging to different users are different, which not only introduces more priori information into the graph structure but also mines specific features of different users.

3) FP and AO-based algorithms' effectiveness and accuracy are demonstrated via convergence and complexity analysis. Simulations and theoretical analysis show that the proposed MHSB-GNN algorithm performs better than the FP-based algorithm with comparable complexity in terms of sum rate for APS&BF design. Moreover, both algorithms have the same order of magnitude of running time.

The rest of the paper is organized as follows. Section II introduces the system model and the problem formulation. Section III provides the details of FP and AO-based algorithms. Section IV describes the proposed MHSB-GNN scheme for the joint APS&BF design. Section V provides the simulation results. Finally, conclusions are drawn in Section VI.

Notations: Lower case letters x denotes scalars. Lower case bold-faced letters $\mathbf{a}$ denote column vectors. Upper case bold-faced letters $\mathbf{A}$ denote matrices. The operators $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ correspond to the transpose and Hermitian transpose, respectively. The real and imaginary parts of a complex number x are denoted by $\Re(x)$ and $\Im(x)$, respectively. $\mathcal{CN}(\cdot)$ denotes complex Gaussian distribution. $\|\cdot\|_p$ denotes the p-

parametrization of one vector and $\mathbb{E}\{\cdot\}$ denotes expectation. $\mathcal{O}(\cdot)$ is the big-$\mathcal{O}$ computational complexity notation. $1j$ is equal to $\sqrt{-1}$. Finally, $\mathbf{0}_L$ denotes the all-zero vector of dimension L.

II. System Model for Cell Free MISO Scenario

A. Network Model

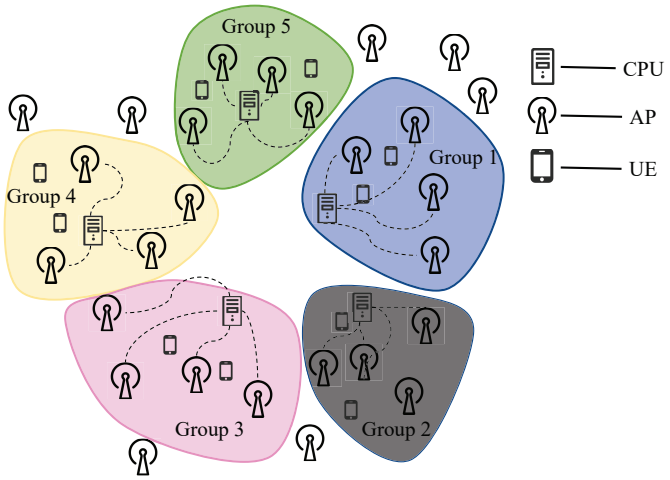


Fig. 1. Cooperation AP clusters for different UE groups in a user-centric cell-free network.

As shown in Fig. 1, we consider a time division duplex (TDD) downlink non-coherent[1] cell-free system, including K APs with N_t antennas and M single-antenna UEs distributed arbitrarily in the scenario. APs are divided into L groups. Each UE selects the AP group relying on the average distance from the UE to the APs in one group. For ease of derivation, we assume that each group has the same number of UEs and APs. Each group has a central processing unit (CPU), which is responsible for the AP cooperation. In each group, different APs send different data streams to one UE, and UEs share the same serving APs. Mathematically, let $\mathcal{M}_l = \{1, \ldots, m, \ldots, M/L\}$ be the UE set of group l and $\mathcal{I}_l = \{1, \ldots, i, \ldots, K/L\}$ be the candidate AP set of group l. Set $\mathcal{L}$ denotes the group set.

The block fading channel between AP i and UE m is denoted as $\mathbf{h}_{mi} = \beta_{mi}\mathbf{g}_{mi}$. The channel power $\beta_{mi} = \Psi_{mi}\mathrm{PL}(d_{mi})$ follows the large-scale fading characteristic describing the shadowing Ψ_{mi} and path loss $\mathrm{PL}(d_{mi})$, where d_{mi} is the 3-dimensional (3D) distance between i-th AP and m-th UE. The elements of small-scale fading $\mathbf{g}_{mi}$ are drawn from independent and identical distribution (i.i.d.) with $\mathcal{CN}(0,1)$.

Let $\mathbf{w}_{mi} \in \mathbb{C}^{1\times N_t}$ be the beamforming vector between i-th AP and m-th UE. Let binary variable $s_i \in \{0,1\}$ be the selecting decision on whether AP i serving UE m or not, $s_i = 1$ indicates AP i serves UE m, and $s_i = 0$, otherwise. Thus, the received signal at UE m is denoted as

$$\begin{aligned} y_m = &\sum\nolimits_{i\in\mathcal{I}_l} s_i \sum\nolimits_{n\in\mathcal{M}_l} \mathbf{h}_{mi}\mathbf{w}_{ni}^{\mathrm{H}} x_{ni} + \\ &\sum\nolimits_{l'\neq l} \sum\nolimits_{j\in\mathcal{I}_{l'}} s_j \sum\nolimits_{n\in\mathcal{M}_{l'}} \mathbf{h}_{mj}\mathbf{w}_{nj}^{\mathrm{H}} x_{nj} + n_m, \end{aligned} \tag{1}$$

[1] Considering non-coherent transmission can avoid the problem of synchronization in CFN [9].

where x_{mi} is the signal and n_m is the additive white Gaussian noise (AWGN), with noise power $\mathbb{E}\{n_m n_m^{\mathrm{H}}\} = \sigma^2$. The right-hand side of (1) includes the desired signal of UE m, inter-group interference, and intra-group interference.

B. Problem Formulation

According to the sum rate formula of non-coherent transmission [40], the achievable data rate at UE $m, m \in \mathcal{M}_l$ in l-th group can be expressed by

$$r_m = \log_2\left(1 + \frac{\mathcal{A}_m}{\mathcal{B}_{1,m} - \mathcal{A}_m + \mathcal{B}_{2,m} + \sigma_m^2}\right), \tag{2}$$

where $\mathcal{A}_m = \sum_{i\in\mathcal{I}_l} s_i \left|\mathbf{h}_{mi}\mathbf{w}_{mi}^{\mathrm{H}}\right|^2, i \in \mathcal{I}_l$, $\mathcal{B}_{1,m} = \sum_{i\in\mathcal{I}_l} s_i \sum_{n\in\mathcal{M}_l}\left|\mathbf{h}_{mi}\mathbf{w}_{ni}^{\mathrm{H}}\right|^2$ and the inter-group interference $\mathcal{B}_{2,m} = \sum_{j\in\mathcal{I}_{l'}} s_j \sum_{n\in\mathcal{M}_{l'}}\left|\mathbf{h}_{mj}\mathbf{w}_{nj}^{\mathrm{H}}\right|^2, l' \neq l$.

This paper aims at maximizing the downlink weighted sum rate (WSR) at the UE side through the cross-layer optimization of beamforming matrix $\mathbf{W} = \{\mathbf{w}_{mi}\}, \forall m \in \mathcal{M}_l, \forall i \in \mathcal{I}_l, l \in \mathcal{L}$ and AP selecting matrix $\mathcal{S} = \{s_i\} \in \{0,1\}^{|\mathcal{I}_l|}, \forall i \in \mathcal{I}_l, l \in \mathcal{L}$.

Mathematically, the problem can be formulated as follows:

$$\begin{aligned} (\mathrm{P0}) \quad \max_{\mathbf{W},\mathcal{S}} \quad & \sum_{l=1}^{L}\sum_{m\in\mathcal{M}_l} \delta_m r_m, \\ \text{s.t.} \quad & s_i \in \{0,1\}, && \text{(3a)} \\ & \sum_{m\in\mathcal{M}_l} \|\mathbf{w}_{mi}\|_2^2 \le \mathrm{P}_{\max}, \forall i \in \mathcal{I}_l, l \in \mathcal{L}, && \text{(3b)} \\ & \sum i \in \mathcal{I}_l s_i \le N, \forall l \in \mathcal{L}, && \text{(3c)} \end{aligned}$$

where δ_m is the prior weight of UE m and $\mathrm{P}_{\max}$ is the maximum transmission power per AP. (3c) limits the number of serving APs for UE m because the ability to decode different streams for one UE is always limited [40].

Intuitively, the optimal solution of (P0) can be obtained with centralized algorithms with global instantaneous CSI. Note that the centralized algorithms result in significant interaction overhead between CPUs. Moreover, the centralized learning-based schemes result in high training overhead for NN, as the large scale of the communication system leads to a significant increase in the size of NN. In order to reduce the overhead and ensure the algorithms applicable to large-scale communication systems, we transform the centralized problem (P0) requiring global CSI into a distributed APS&BF design problem requiring local instantaneous CSI.

First, we propose a two-timescale frame structure, as shown in Fig. 2. The time is divided into several frames, and each frame consists of T_s time slots. The channel statistics, including UEs' location, remains unchanged during several frames [41]. Based on this premise, the statistical interference CSI remains constant throughout the frame and is exchanged by different groups at the beginning of each frame τ. The local instantaneous CSI is obtained at the beginning of each time slot. For UE $m, m \in \mathcal{M}_l$ in the current l-th group, the local instantaneous CSI is the channel from the $i, i \in \mathcal{I}_l$-th AP to the $m, m \in \mathcal{M}_l$-th UE of the current l-th group, denoted as $\mathbf{h}_{mi}$.

The statistical interfering CSI is the statistical channel from the $j, j \in \mathcal{I}_{l'}$-th AP of the $l', l' \neq l$-th group to the m-th UE from the l-th group, denoted as $\mathbf{R}_{mj} = \mathbb{E}\left\{\mathbf{h}_{mj}^{\mathrm{H}}\mathbf{h}_{mj}\right\} \in \mathbb{C}^{N_t \times N_t}$.

We impose an extra effective power constraint $\sum_{n\in\mathcal{M}_{l'}} \mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{w}_{nj}^{\mathrm{H}}\right) \leq \mathrm{P}_j, j \in \mathcal{I}_{l'}$ on each AP. This extra power constraint is exactly the per-AP power constraint (3b) in (P0). In this case, imposing an extra power constraint will not lose optimality. With the above power constraint, we derive a lower bound $\mathring{r}_m$ for the actual data rate r_m by replacing the inter-group interference $\mathcal{B}_{2,m}$ with an interference upper bound in (2). The lower bound $\mathring{r}_m$ only depends on the channel statistics of the cross-links and the local CSI within group l. Then the following lemma is derived.

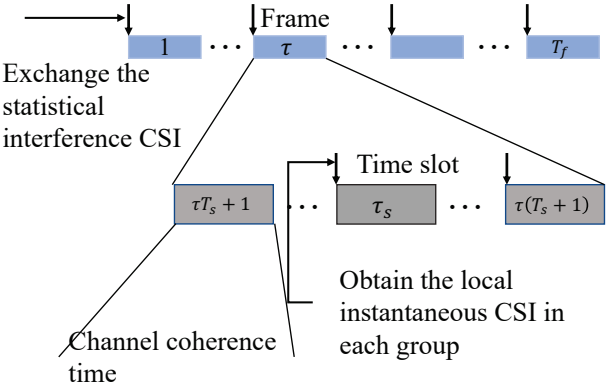


Fig. 2. An illustration of two-timescale frame structure to obtain instantaneous partial CSI and outdated CSI.

Lemma 1: (Lower Bound of Data Rate) Suppose that the upper bound of each AP power is $\sum_{n\in\mathcal{M}_{l'}} \mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{w}_{nj}^{\mathrm{H}}\right) \leq \mathrm{P}_j$. A lower bound of the achievable rate of user m is given by

$$\mathring{r}_m = \log\left(1 + \frac{\mathcal{A}_m}{\mathcal{B}_{1,m} - \mathcal{A}_m + \overline{\mathcal{B}}_{2,m} + \sigma_m^2}\right), \tag{4}$$

and

$$\overline{\mathcal{B}}_{2,m} = \sum\nolimits_{j\in\mathcal{I}_{l'}} s_j \sum\nolimits_{n\in\mathcal{M}_{l'}} \mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{R}_{mj}\mathbf{w}_{nj}^{\mathrm{H}}\right), \tag{5}$$

where $\mathbf{R}_{mj} = \mathbb{E}\left\{\mathbf{h}_{mj}^{\mathrm{H}}\mathbf{h}_{mj}\right\} \in \mathbb{C}^{N_t\times N_t}$ is the covariance matrix of the interfering channel $\mathbf{h}_{mj}$.

Proof: Please refer to Appendix A for the proof. ■

By replacing the original optimization function with the lower bound of the sum rate, the centralized modeled problem (P0) can be approximated by the distributed modeled problem in l-th group as follows,

$$\begin{aligned}(\mathrm{P1}) \quad \max_{\mathbf{W},\mathcal{S}} \quad & \sum\nolimits_{m\in\mathcal{M}_l} \delta_m \mathring{r}_m, \\ \text{s.t.} \quad & (3a), && (6a)\\ & \sum\nolimits_{m\in\mathcal{M}_l} \|\mathbf{w}_{mi}\|_2^2 \leq \mathrm{P}_{\max}, \forall i \in \mathcal{I}_l, && (6b)\\ & \sum\nolimits_{i\in\mathcal{I}_l} s_i \leq N. && (6c)\end{aligned}$$

Simulations results in Fig. 3 also show that the gap between the "Actual Data Rate" obtained by solving (P0) and the "Best Data Rate" obtained by solving (P1) is narrow. The detailed simulation setup is given in Section V. Therefore, The lower rate bound $\mathring{r}_m$ in Lemma 1 is a good approximation of the actual rate r_m.

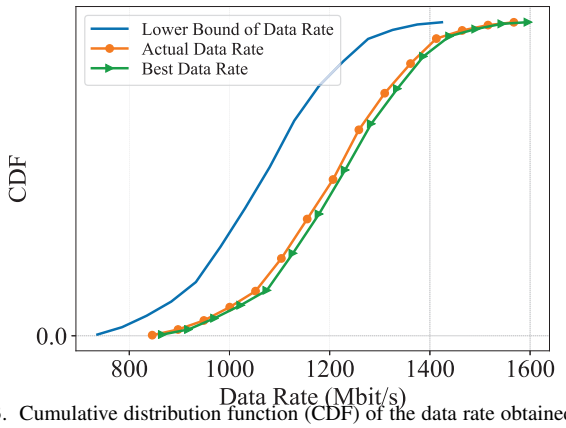


Fig. 3. Cumulative distribution function (CDF) of the data rate obtained from 150 channel samples with AO-based alg.

Note that the approximate data rate $\mathring{r}_m$ only depends on the local channels $\mathbf{h}_{mi}, i \in \mathcal{I}_l$ and the statistical interference channels $\mathbf{R}_{mj} = \mathbb{E}\left\{\mathbf{h}_{mj}^{\mathrm{H}}\mathbf{h}_{mj}\right\}, j \in \mathcal{I}_{l'}$. Solving (P1) is challenging as it is a NP-problem. The non-linear fractional structure and the binary variable s_i make (P1) a mixed-integer non-linear programming problem. Additionally, from (6b) and (6c), different s_i and $\mathbf{w}_{mi}$ for UE m strongly influences the AP selection and beamforming design for other UEs.

C. Problem Transformation

Assume $\mathbf{W}_i = [\mathbf{w}_{1i}, \ldots \mathbf{w}_{mi}, \ldots], m \in \mathcal{M}_l$. Without loss of generation, if AP i serves UE m, i.e., $\mathbf{W}_i$ must exist and $s_i = s_{mi} = 1$ holds. $s_i = s_{mi}$ holds as the UEs in one group select the same AP set. Naturally, the relationship between s_i and $\mathbf{W}_i$ is constructed as $s_i = \left\|\|\mathbf{W}_i\|_2^2\right\|_0$ [13]. However, the equivalent problem is still challenging because of the discontinues l_0-norm. The reweighted l_1-norm is a smooth function that can approximate l_0-norm. Assume a vector $\mathbf{x} = [x_1, \ldots, x_i, \ldots, x_I], x_i \in \mathrm{R}$ and a diagonal matrix $\alpha = \mathrm{diag}\{\alpha_1, \ldots, \alpha_i, \ldots, \alpha_I\}, \alpha_i > 0$. The l_0-norm of $\mathbf{x}$ is approximated by the reweighted l_1-norm of $\mathbf{x}$ as $\|\mathbf{x}\|_0 \approx \sum_i \alpha_i |x_i|$ [19]. Based on the above approximation operation and $\|\mathbf{W}_i\|_2^2$ is a scalar. Then we get $\left\|\|\mathbf{W}_i\|_2^2\right\|_0 \approx \alpha_i \|\mathbf{W}_i\|_2^2$, where α_i is computed as

$$\alpha_i = \frac{1}{\|\mathbf{W}_i\|_2^2 + \epsilon} = \frac{1}{\sum_{m\in\mathcal{M}_l}\|\mathbf{w}_{mi}\|_2^2 + \epsilon}, \tag{7}$$

where $\epsilon > 0$ keeps α_i away from being infinity. The smaller ϵ, the better the approximation of l_1-norm to l_0-norm. Usually ϵ is set to 10^{-6}, which results in negligible error on the achievable data rate. After relaxation of the binary variable s_i, Problem (P1) is reformulated as

$$\begin{aligned}(\mathrm{P2}) \quad \max_{\mathbf{W}} \quad & f_o(\mathbf{W}), \\ \text{s.t.} \quad & (6b), && (8a)\\ & \sum\nolimits_{i\in\mathcal{I}_l} \alpha_i \|\mathbf{W}_i\|_2^2 \leq N, && (8b)\end{aligned}$$

with

$$f_o = \sum_{m\in\mathcal{M}_l} \delta_m \log\left(1+\frac{\mathcal{A}_m^{\mathrm{o}}}{\mathcal{B}_{1,m}^{\mathrm{o}}-\mathcal{A}_m^{\mathrm{o}}+\mathcal{B}_{2,m}^{\mathrm{o}}+\sigma_m^2}\right), \quad (9)$$

where $\mathcal{A}_m^{\mathrm{o}} = \sum_{i\in\mathcal{I}_l}\left|\mathbf{h}_{mi}\mathbf{w}_{mi}^{\mathrm{H}}\right|^2$, $\mathcal{B}_{1,m}^{\mathrm{o}} = \sum_{i\in\mathcal{I}_l}\sum_{n\neq m,n\in\mathcal{M}_l}\left|\mathbf{h}_{mi}\mathbf{w}_{ni}^{\mathrm{H}}\right|^2$, and $\mathcal{B}_{2,m}^{\mathrm{o}} = \sum_{j\in\mathcal{I}_{l'}}\sum_{n\in\mathcal{M}_{l'}}\mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{R}_{mj}\mathbf{w}_{nj}^{\mathrm{H}}\right)$.

III. Joint Optimization of AP Selection and Beamforming

This section proposes AO and FP-based algorithms to solve the transformed problem (P2) . Then, these algorithms' optimality, convergence, and complexity analysis are also provided.

A. Direct Solution

First, the AO-based alg[2] is introduced. After transforming the MINP problem into a general continuous problem, the multi-dimensional quadratic transform (Theorem 2 in [23]) is introduced to directly transform each SINR term in $f_o(\mathbf{W})$ into the sum form. Then the problem (P2) is rewritten as follows,

$$\begin{aligned}(\mathrm{P3}) \quad &\max_{\mathbf{W},\mathbf{Y}} \ f_d(\mathbf{W},\mathbf{Y}), &(10a)\\ &\text{s.t.} \quad (6b),(8b)\end{aligned}$$

where f_{d} is the new objective of problem (P2), as shown in (11) with $\mathcal{B}_m^{\mathrm{o}} = \mathcal{B}_{1,m}^{\mathrm{o}} + \mathcal{B}_{2,m}^{\mathrm{o}}$.

An auxiliary variable $\mathbf{y}_{mi}$ is introduced concerning each pair, and the collection of auxiliary variable $\{\mathbf{y}_{mi}\}, m \in \mathcal{M}_l, i\in\mathcal{I}_l$ is denoted by $\mathbf{Y}$. When $\mathbf{Y}$ is fixed, the problem (P3) is convex about $\mathbf{w}_{mi}$. $\mathbf{y}_{mi}$ and $\mathbf{w}_{mi}$ can be optimized in an iterative form. The optimal $\mathbf{y}_{mi}$ for fixed $\mathbf{w}_{mi}$ can be computed by $\partial f_d/\partial\mathbf{y}_{mi}=0$. The result is

$$\mathbf{y}_{mi}^* = \left(\sigma_m^2+\mathcal{B}_m^{\mathrm{o}}-\mathcal{A}_m^{\mathrm{o}}\right)^{-1}\mathbf{w}_{mi}\mathbf{h}_{mi}^{\mathrm{H}}. \quad (13)$$

Then, the rest of the optimization problem for finding the optimal $\mathbf{w}_{mi}$ under fixed $\mathbf{y}_{mi}$ is convex, which can be solved by the interior-point method [42]. By alternately optimizing $\mathbf{y}_{mi}$ and $\mathbf{w}_{mi}$, a local optimum of (P3) can be obtained. This approach is referred as the alternating optimization (AO)-based algorithm. The AO-based alg is outlined in Algorithm 1.

[2] For ease of presentation, "Alg" and "alg" stand for "Algorithm" and "algorithm" in the paper.

Algorithm 1 Iterative AO-based algorithm for APS&BF

Input: Initialize α, $\mathbf{W}$ and $\mathcal{S}$ to feasible values,

1: t: Iteration Index, starting with $t=1$.
2: ε: The maximum tolerance, $\varepsilon>0$.
3: **while** $\left|f_o(\mathbf{W})^{(t+1)} - f_o(\mathbf{W})^{(t)}\right| > \varepsilon$ **do**
4: update $\mathbf{Y}^{(t+1)}$ according to (13) with $\mathbf{W}^{(t)}$,
5: update $\mathbf{W}^{(t+1)}$ by solving problem (P3) over fixed $\mathbf{Y}^{(t+1)}$ with MATLAB toolbox CVX,
6: update $\alpha^{(t+1)}$ according to (7) with $\mathbf{W}^{(t+1)}$,
7: set $t=t+1$,
8: **end while**

Output: Beamforming Matrix $\mathbf{W}$ and selecting matrix $\mathcal{S}$.

B. Closed Form Solution

As the complexity of AO-based alg shows exponential growth with the number of users and APs, we introduce a low-complexity scheme, FP-based alg. Each iteration of FP-based alg is performed in closed form rather than numerically solving a convex optimization problem. Lagrange dual transform [23] [24] is introduced to convert the sum logarithm structure in $f_o(\mathbf{W})$ into the sum of ratio form $f_{\boldsymbol{\gamma}}(\boldsymbol{\gamma},\mathbf{W})$ by introducing multipliers $\boldsymbol{\gamma}=\{\gamma_m\}, \forall m\in\{1,\ldots,M\}$, as shown in (12).

When $\mathbf{w}_{mi}$ is fixed, $f_{\boldsymbol{\gamma}}(\boldsymbol{\gamma},\mathbf{W})$ is convex about multiplier γ_m, and optimal γ_m can be computed through setting $\partial f_{\boldsymbol{\gamma}}/\partial\gamma_m=0$. Therefore, the optimal γ_m is as follows,

$$\gamma_m^* = \frac{\mathcal{A}_m^{\mathrm{o}}}{\mathcal{B}_m^{\mathrm{o}}-\mathcal{A}_m^{\mathrm{o}}+\sigma_m^2}. \quad (14)$$

After $\boldsymbol{\gamma}$ is solved, there is only one variable left. (12) is a multi-dimensional sum-of-ratio form, which can be reconstructed by a multi-dimensional quadratic transform [23]. By treating $(1+\gamma_m)\mathcal{A}_m^{\mathrm{o}}$ as the numerator and $\mathcal{B}_m^{\mathrm{o}}+\sigma_m^2$ as the denominator, respectively, (12) is rewritten as $f_q(\boldsymbol{\gamma},\mathbf{W},\mathbf{Y})$,

$$f_q = \sum_{m\in\mathcal{M}_l}\delta_m\left[\log(1+\gamma_m)-\gamma_m\right] + f_{\mathrm{p}}(\boldsymbol{\gamma},\mathbf{W},\mathbf{Y}), \quad (15)$$

with $f_{\mathrm{p}}(\boldsymbol{\gamma},\mathbf{W},\mathbf{Y})$,

$$\begin{aligned}f_{\mathrm{p}} = &\sum_{m\in\mathcal{M}_l} 2\sqrt{\delta_m(1+\gamma_m)}\sum_{i\in\mathcal{I}_l}\Re\left\{\mathbf{y}_{mi}^{\mathrm{H}}\mathbf{w}_{mi}\mathbf{h}_{mi}^{\mathrm{H}}\right\} - \\ &\sum_{m\in\mathcal{M}_l}\sum_{i\in\mathcal{I}_l}\left|\mathbf{y}_{mi}\right|^2\left(\sigma_m^2+\mathcal{B}_m^{\mathrm{o}}\right),\end{aligned} \quad (16)$$

where an auxiliary variable $\mathbf{y}_{mi}$ is introduced with respect to each pair, and the collection of auxiliary variable $\{\mathbf{y}_{mi}\}, \forall m\in\mathcal{M}_l, \forall i\in\mathcal{I}_l$ is denoted by $\mathbf{Y}$.

$$f_{\mathrm{d}} = \sum_{m\in\mathcal{M}_l}\delta_m\log\left(1+2\sum_{i\in\mathcal{I}_l}\Re\left\{\mathbf{y}_{mi}^{\mathrm{H}}\mathbf{w}_{mi}\mathbf{h}_{mi}^{\mathrm{H}}\right\} - \sum_{i\in\mathcal{I}_l}\left|\mathbf{y}_{mi}\right|^2\left(\sigma_m^2+\mathcal{B}_m^{\mathrm{o}}-\mathcal{A}_m^{\mathrm{o}}\right)\right). \quad (11)$$

$$f_{\boldsymbol{\gamma}}(\boldsymbol{\gamma},\mathbf{W}) = \sum\nolimits_{m\in\mathcal{M}_l}\delta_m\left[\log_2(1+\gamma_m)-\gamma_m\right] + \sum\nolimits_{m\in\mathcal{M}_l}\delta_m(1+\gamma_m)\frac{\mathcal{A}_m^{\mathrm{o}}}{\mathcal{B}_m^{\mathrm{o}}+\sigma_m^2}. \quad (12)$$

Thus, the objective function is transformed from f_γ to f_q. Then, the problem (P2) is reformulated as follows,

$$\text{(P4)} \quad \max_{\boldsymbol{\gamma},\mathbf{W},\mathbf{Y}} \; f_q\left(\boldsymbol{\gamma},\mathbf{W},\mathbf{Y}\right), \tag{17a}$$
$$\text{s.t.} \quad \text{(6b)}, \text{(8b)}$$

After the update of $\boldsymbol{\gamma}$ in (14), the left two variables, $\mathbf{Y}$ and $\mathbf{W}$, need to be optimized. Firstly, we assume other variables are fixed except $\mathbf{Y}$, which can be explicitly determined through solving $\partial f_q/\partial \mathbf{y}_{mi} = 0$. Therefore, the optimal $\mathbf{y}_{mi}$ is computed as follows,

$$\mathbf{y}_{mi}^{*} = \left(\sigma_m^2 + \mathcal{B}_m^{\mathrm{o}}\right)^{-1} \sqrt{\delta_m(1+\gamma_m)}\mathbf{h}_{mi}\mathbf{w}_{mi}^{\mathrm{H}}. \tag{18}$$

Until now, only one variable $\mathbf{W}$ is left. When the other two variables are fixed, the reconstructed objective $f_q\left(\boldsymbol{\gamma},\mathbf{W},\mathbf{Y}\right)$ with constraints (3b) (8b) about $\mathbf{w}_{mi}$ is a classic convex quadratic optimization problem. Unlike the AO-based alg, $\mathbf{w}_{mi}$ is solved by an efficient closed-form method. The Lagrangian form of (15) is derived by introducing two Lagrangian multipliers, $\eta_i \geq 0$ and $\lambda \geq 0$, related to per AP power constraint and number of receiving data streams constraint, respectively. Then the final form of f_q is $\mathcal{L}_{f_q}\left(\boldsymbol{\gamma},\mathbf{W},\mathbf{Y}\right)$,

$$\begin{aligned}\mathcal{L}_{f_q} = &\sum_{m\in\mathcal{M}_l} \delta_m\left[\log(1+\gamma_m)-\gamma_m\right] + \sum_{m\in\mathcal{M}_l} f_{\mathrm{p}}\left(\mathbf{w}_{mi}\right) \\ &- \sum_{i\in\mathcal{I}_l}\eta_i\left(\sum_{m=1}^{M}\left\|\mathbf{w}_{mi}\right\|_2^2 - \mathrm{P}_{\max}\right) \\ &- \lambda\left(\sum_{i\in\mathcal{I}_m}\alpha_i\sum_{m\in\mathcal{M}_l}\left\|\mathbf{w}_{mi}\right\|_2^2 - N\right).\end{aligned} \tag{19}$$

Let $\partial\mathcal{L}_{f_q}/\partial\mathbf{w}_{mi} = 0$, the optimal $\mathbf{w}_{mi}$ can be derived as follows,

$$\begin{aligned}\mathbf{w}_{mi}^{*} = &\left(\sum_{n\in\mathcal{M}_l}\sum_{j\in\mathcal{I}_l}\mathbf{h}_{ni}^{H}\mathbf{y}_{nj}\mathbf{y}_{nj}^{H}\mathbf{h}_{ni} + (\eta_i+\lambda\alpha_i)\mathbf{I}\right)^{-1} \\ &\sqrt{\delta_m\left(1+\gamma_m\right)}\left(\mathbf{h}_{mi}\mathbf{y}_{mi}^{H}\right).\end{aligned} \tag{20}$$

Note that multipliers η_i and λ need to be solved, and these two constraints cannot be tight simultaneously. According to complementary slackness, one of these two multipliers must be zero. Hence, at each iteration, a heuristic algorithm is designed, where if constraint (6b) can be satisfied, $\eta_i = 0$; otherwise, λ is initialized with a non-zero value, and η_i is determined by bisection search to meet (6b). The proposed FP-based algorithm is outlined in Algorithm 2.

C. The Optimality, Convergence and Complexity of AO-based and FP-based Algorithms

The two proposed traditional schemes can only guarantee a high-quality suboptimal solution to the original problem (P1). The reasons are as follows. First, to transform the original problem (P1) to the more tractable problem (P2), we relax the binary variables s_i by approximating the l_0-norm with the l_1-norm, thus leading to a performance loss. Second, for the AO-based alg, the transformed problem (P2) is solved by applying the quadratic transformation and CVX techniques in the alternating optimization framework. Thus, only a locally optimal solution can be guaranteed. Third, for the FP-based alg, the transformed problem (P2) is also solved in the alternating optimization framework. Unlike AO-based alg, FP-based alg is solved by introducing Lagrange dual transform and quadratic transform techniques, which guarantees a locally optimal solution. Generally, both AO-based and FP-based algorithms are high-quality suboptimal solutions to the original problem (P1).

Algorithm 2 Iterative Closed Form FP-based algorithm for APS&BF

Input: Initialize $\mathbf{W}$ and $\mathcal{S}$ to feasible values,

1: t: Iteration index, starting with $t = 1$.
2: ε: The maximum tolerance, $\varepsilon > 0$.
3: **while** $\left|f_o\left(\mathbf{W}\right)^{(t+1)} - f_o\left(\mathbf{W}\right)^{(t)}\right| > \varepsilon$ **do**
4: compute $\boldsymbol{\gamma}^{(t+1)}$ according to (14) with $\mathbf{W}^{(t)}$,
5: compute $\mathbf{Y}^{(t+1)}$ according to (18) with $\boldsymbol{\gamma}^{(t+1)}$ and $\mathbf{W}^{(t)}$,
6: compute $\mathbf{W}^{(t+1)}$ according to (20) with $\boldsymbol{\gamma}^{(t+1)}$ and $\mathbf{Y}^{(t+1)}$,
7: determine $\left\{\eta_i^{(t+1)}, \lambda^{(t+1)}, \forall m \in \mathcal{M}\right\}$ through the heuristic algorithm, as described in the paragraph after (20),
8: update $\alpha^{(t+1)}$ according to (7) with $\mathbf{W}^{(t+1)}$,
9: set $t = t + 1$,
10: **end while**

Output: Beamforming matrix $\mathbf{W}$ and selecting matrix $\mathcal{S}$.

1) **AO-based alg**: The AO-based algorithm is essentially a block coordinate ascent algorithm for the reformulated problem (P3), which is a convex optimization problem due to the concave-convex form of (P2), so it converges to a stationary point $(\mathbf{w}_{mi}^{*}, \mathbf{Y}^{*})$ of (P3). According to C3 and C2 in [23], the first-order condition on $\mathbf{w}_{mi}$ for (P3) under the optimal $\mathbf{Y}^{*}$ is the same as for the original problem (P1). Hence, the algorithm also converges to a stationary point of (P1). Condition C3 in [23] guarantees that the sum-of-functions-of-ratio value is nondecreasing after every updating of $\mathbf{Y}^{*}$.

The main complexity of the AO-based algorithm lies in steps 4 and 5 in Algorithm 1. Assume T_{iter} is the number of iterations of alg. Specifically, the complexity of step 4 is $\mathcal{O}\left(\left|\mathcal{M}_l\right|^2\left|\mathcal{I}_l\right|^2 N_t\right)$. In step 5, (P3) can be solved by the interior-point method, whose complexity is $\mathcal{O}\left(N_t^{3.5}\left|\mathcal{M}_l\right|\left|\mathcal{I}_l\right|\right)$ [42]. Therefore, the total complexity of AO-based alg is $\mathcal{O}\left(T_{\mathrm{iter}}\left(N_t^{3.5}\left|\mathcal{M}_l\right|\left|\mathcal{I}_l\right| + \left|\mathcal{M}_l\right|^2\left|\mathcal{I}_l\right|^2 N_t\right)\right)$.

2) **FP-based alg**: The FP-based algorithm is non-decreasing in the objective function (12) within each iteration. The non-decreasing convergence can be proven by considering the following chain of reasoning going from iteration t to $t+1$:

$$f_o(\mathbf{W}^{(t)}) \overset{(a)}{=} f_{\boldsymbol{\gamma}}(\mathbf{W}^{(t)}, \boldsymbol{\gamma}^{(t)}) \overset{(b)}{\leq} f_{\boldsymbol{\gamma}}(\mathbf{W}^{(t)}, \boldsymbol{\gamma}^{(t+1)}) \tag{21a}$$
$$\overset{(c)}{=} f_q(\mathbf{W}^{(t)}, \boldsymbol{\gamma}^{(t+1)}, \mathbf{Y}^{(t)}) \tag{21b}$$

$$\overset{(d)}{\leq} f_q(\mathbf{W}^{(t)}, \boldsymbol{\gamma}^{(t+1)}, \mathbf{Y}^{(t+1)}) \tag{21c}$$

$$\overset{(e)}{\leq} f_q(\mathbf{W}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \mathbf{Y}^{(t+1)}) \tag{21d}$$

$$\overset{(f)}{=} f_{\boldsymbol{\gamma}}(\mathbf{W}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}) \overset{(g)}{=} f_o(\mathbf{W}^{(t+1)}), \tag{21e}$$

where (a) holds because the reformulated objective function $f_{\boldsymbol{\gamma}}$ equals to the original function f_o when the optimal $\boldsymbol{\gamma}$ is substituted; (b) derives from the fact that the update of $\boldsymbol{\gamma}$ with other fixed variables maximizes $f_{\boldsymbol{\gamma}}$; (c) follows the similar step as (a); From the fact that the update of $\mathbf{Y}$ or $\mathbf{W}$ with other fixed variables maximizes f_q, (d) or (e) can be derived; Both (f) and (g) follow the similar steps as (c) and (a), respectively. So far, the objective function f_o has been proven non-decreasing because the function f_o is bounded. Thus, the FP-based algorithm converges, and the original objective function f_o arrives at a local optimum.

The main computational complexity of the FP-based algorithm lies in steps 4-7 in Algorithm 2. Specifically, the computational complexity of steps 4-6 is $\mathcal{O}\left(\left|\mathcal{M}_l\right|^2 \left|\mathcal{I}_l\right|^2 N_t\right)$. The complexity of computing the neighboring interference is $\mathcal{O}\left(\left|\mathcal{M}_l\right| \left|\mathcal{I}_l\right| L\right)$. In step 7, the complexity of searching for Lagrangian multiplier η_i is $\mathcal{O}\left(\log_2 \theta\right)$. Therefore, the total complexity of the FP-based algorithm is $\mathcal{O}\left(T_{\text{iter}}\left(\left|\mathcal{M}_l\right|^2 \left|\mathcal{I}_l\right|^2 N_t + \left|\mathcal{I}_l\right| \log_2 \theta\right) + \left|\mathcal{M}_l\right| \left|\mathcal{I}_l\right| L\right)$.

IV. Graph Neural Network Architecture Design

These two conventional algorithms require iterative optimization, especially AO-based alg, which incurs a high computational complexity. Although FP-based alg has low computational complexity, it is inferior to AO-based alg in terms of WSR performance. This section proposes a machine learning-based scheme where each group applies APS&BF alg independently with local CSI and statistical interference CSI. In the following subsections, the graph representation will be introduced first. Then, the structure of the proposed distributed machine learning algorithm, and network training will be elaborated. Finally, the equivalence between GNN and FP-based algorithm is proved.

A. Graph Presentation for Cross-Layer Optimization Problem

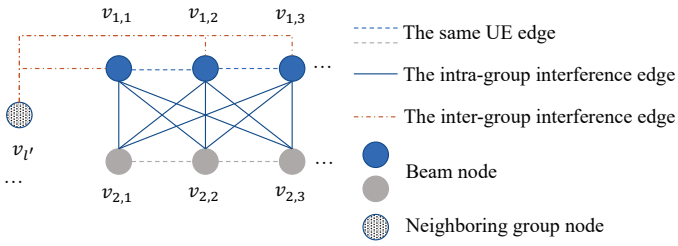


Fig. 4. Graph Modeling of Cell Free Networks, $G = \{V_1, V_2, E_1, E_2, E_3\}$.

The proposed MHSB-GNN-based beamforming design aims to map the local instantaneous CSI, the historical interference CSI, the transmit power, and the interference power to the beamforming vectors. The GNN in previous research is responsible for the single-cell single-AP-to-multi-UE beamforming design, where UEs and APs are modeled as nodes [33], [43]. This paper aims to solve the single-cell multi-AP-to-multi-UE beamforming problem, which differs from the previous research. We innovatively model the beams as nodes and other neighboring groups as another kind of nodes, providing an intuitive representation of the beam-to-beam relationship. This graph representation is suitable for single-cell multi-AP-to-multi-UE beamforming design, which helps the algorithm's distributed implementation and makes it more suitable for large-scale scenarios.

As Fig. 4 shows, we model the current group l as a graph $G = \{V_1, V_2, E_1, E_2, E_3\}$. There are $|V_1|$ beam nodes denoted as $V_1 = \{v_{m,i} | m \in \mathcal{M}_l, i \in \mathcal{I}_l\}$, representing the transmission of each beam for each UE. The beam node feature vector is denoted as $\mathbf{z}_{mi} \in \mathbb{R}^{1\times 2N_t}$. The beam node representation vector is denoted as $\mathbf{x}_{mi} \in \mathbb{R}^{1\times 2N_t}$. We also introduce another $|V_2| = L - 1$ neighboring group nodes, $V_2 = \left\{v_{l'} | l' \neq l\right\}$, whose feature vector is denoted as $\mathbf{z}_{l'} \in \mathbb{R}^{1\times 1}$. If two beam nodes belong to one user, there is no intra-group interference edge between them; otherwise, there is a bidirectional intra-group interference edge $\mathbf{a}_{mi,nj}, n \neq m$ between them. The edges $\mathbf{b}_{mi,l'}$ from the neighboring group nodes to beam nodes are inter-group interference edges. The beam nodes contain the representation vectors to be updated, which is the output of MHSB-GNN. The neighboring group nodes do not have representation vectors to be updated, and they provide information for updating the beam nodes. Modeling the neighboring group nodes without representation vectors can reduce the computational size of the GNN and also save the training overhead of the GNN. Such graph representation makes GNN a lightweight network and more suitable for large-scale systems.

B. Structure Design for Multi-Head Single-Body GNN

In this subsection, we will introduce the architecture of the proposed MHSB-GNN in detail. MHSB-GNN aims to encode all valuable information of corresponding nodes in these representation vectors $\mathbf{x}_{mi}$. The overall network consists of one initial module, T_{NN} node updating modules and one normalization module, as illustrated in Fig. 5 (on the next page) and described as follows,

1) **Initial module**: The inputs of the initial module include the current group CSI, the statistical interference CSI, AP transmitting power, and noise power. Specifically, the inputs are beam node feature $\mathbf{z}_{mi} \in \mathbb{R}^{1\times 2N_t}, i \in \mathcal{S}_l, m \in \mathcal{M}_l$, neighboring group node feature $\mathbf{z}_{l'} \in \mathbb{R}^{1\times 1}, l' \neq l$, edge features $\mathbf{a}_{mi,nj}, n \neq m, j \neq i$, and $\mathbf{b}_{mi,l'}, l' \neq l$. To compute the data rate easily during the training phase, we simply separate the channel into the real and imaginary parts, denoted as $\mathbf{z}_{mi} = \text{vec}\left[(\Re\{\mathbf{h}_{mi}\})^{\text{T}}, (\Im\{\mathbf{h}_{mi}\})^{\text{T}}\right]$, $\mathbf{a}_{mi,nj} = \text{vec}\left[(\Re\{\mathbf{h}_{mj}\})^{\text{T}}, (\Im\{\mathbf{h}_{mj}\})^{\text{T}}\right]$, $\mathbf{z}_{l'} = \text{P}_{\max}$, and $\mathbf{b}_{mi,l'} = [\ldots, \text{vec}\left(\Re\left\{\mathbf{R}_{mj}\right\}, \Im\left\{\mathbf{R}_{mj}\right\}\right), \ldots], j \in \mathcal{I}_{l'}$.

For each beam node $v_{m,i}$, the node feature $\mathbf{z}_{mi}$ is processed by a shared initial network, denoted as f_{init}, expressed as $\mathbf{x}_{mi}^{(0)} = f_{\text{init}}\left(\mathbf{z}_{mi}\right)$. Similarly, for the intra-group interfering edge, the edge feature $\mathbf{a}_{mi,nj}$ is processed by the same initial network, expressed as $\mathbf{e}_{mi,nj} = f_{\text{init}}\left(\mathbf{a}_{mi,nj}\right)$. For the inter-group interfering edge, the node feature and edge feature

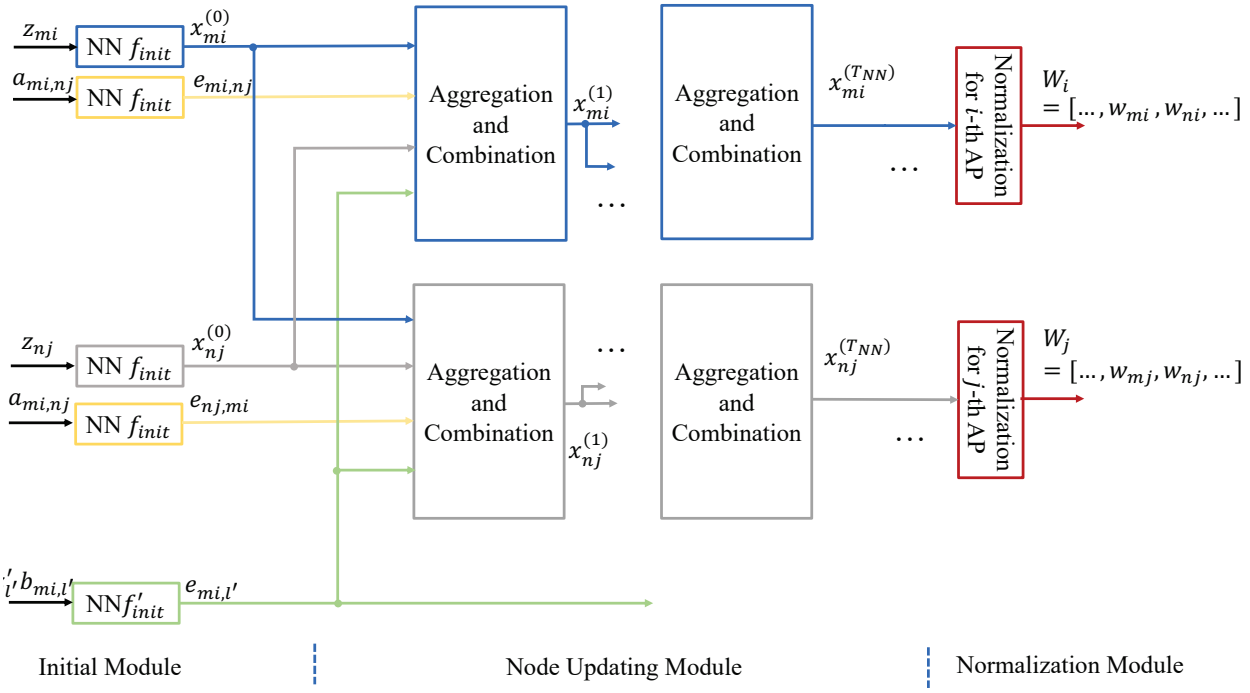


Fig. 5. Overall GNN architecture with one initialization module, T_{NN} node updating modules, and one normalization module.

$\mathbf{z}_{l'}\,\mathbf{b}_{mi,l'}$ are processed by another initial network, denoted as f'_{init}, expressed as $\mathbf{e}_{mi,l'} = f'_{\mathrm{init}}\left(\mathrm{P}_{\max}\mathbf{b}_{mi,l'}\right)$, where the input dim of f'_{init} is different from that of f_{init} and the output dim of f'_{init} is equal to that of f_{init}.

2) **Node updating module**: The initialized feature vectors are passed to the T_{NN} node updating modules to update the representation vectors $\mathbf{x}_{mi}$. T_{NN} is a hyper-parameter. Each node updating module contains two blocks: the aggregation and combination blocks. The update of the representing vector $\mathbf{x}_{mi}^{(t)}, t = 1, \dots, T_{\mathrm{NN}}$ at the t-th module is based on combining its previous representation $\mathbf{x}_{mi}^{(t-1)}$, the aggregation of the previous presentations $\mathbf{x}_{nj}^{(t-1)}, (n,j) \neq (m,i)$ of its neighboring nodes, the aggregation of its neighboring edges features $\mathbf{e}_{mi,nj}, n \neq m$, and the aggregation of the features $\mathbf{e}_{mi,l'}, l' \neq l$ from the neighboring group nodes V_2.

One key in designing GNN is to design an effective node updating module so that the GNN can achieve comparable performance. A multi-head single-body (MHSB) structure is designed to serve as the node updating module. Each aggregation block has two branches, namely neighboring node feature aggregation branch $f_{m,\mathrm{node}}^{(t)}$ and edge feature aggregation branch $f_{m,\mathrm{edge}}^{(t)}$. The combination block is denoted as $f_{m,\mathrm{com}}^{(t)}$. Note that the aggregation and combination blocks corresponding to different UEs are different, while the beam nodes belonging to the same UE share the same aggregation and combination blocks. Using different blocks for different UEs helps mine the different features of UEs. The different features of UEs mean the distance information and the interference information. While for the nodes of the same UE, the shared neural network can exploit the similarity between beams. We refer to this kind of structure as multi-head single-body structure. Another key in designing GNN is to choose a suitable pooling function so that the GNN is scalable and generalizes well. We choose the element-wise mean function, represented by mean_{nj}, as the pooling function PL_{nj} because it is invariant to the permutation of the inputs. The aggregation blocks are described as follows.

Fig. 6 demonstrates an efficient implementation of the aggregation block in MHSB-GNN for node v_{mi}, which consists of two components as described in (22) and (23). In (22) and (23), $\mathbf{x}_{mi}^{(t-1)}$, $\mathbf{N}_{mi}^{(t)}$ and $\mathbf{E}_{mi}^{(t)}$ represent the representation vector of the mi-th node in the $t-1$-th module, the embedding vectors

$$\mathbf{N}_{mi}^{(t)} = \mathrm{PL}_{nj}\left(f_{m,\mathrm{node}}^{(t)}\left(\mathbf{x}_{nj}^{(t)}, |\mathcal{W}_{m,1}^{(t)}\right)\right) = \mathrm{mean}_{nj}\left(f_{m,\mathrm{node}}^{(t)}\left(\mathbf{x}_{nj}^{(t-1)}, |\mathcal{W}_{m,1}^{(t)}\right)\right), n \neq m, n \in \mathcal{M}_l, j \in \mathcal{I}_l \tag{22}$$

$$\mathbf{E}_{mi}^{(t)} = \mathrm{PL}_{nj}\left(f_{m,\mathrm{edge}}^{(t)}\left(\mathbf{b}_{mi,nj}|\mathcal{W}_{m,2}^{(t)}\right)\right) = \mathrm{mean}_{nj}\left(f_{m,\mathrm{edge}}^{(t)}\left(\mathbf{b}_{mi,nj}|\mathcal{W}_{m,2}^{(t)}\right)\right), n \neq m, n \in \mathcal{M}_l, j \in \mathcal{I}_l \tag{23}$$

$$\mathbf{w}_{mi} = \begin{cases} \frac{\mathrm{P}_{\max}}{X_{\mathrm{sum}}}\left(\mathbf{x}_{mi}^{(T_{\mathrm{NN}})}[1:Nt] + 1j \times \mathbf{x}_{mi}^{(T_{\mathrm{NN}})}[Nt+1:2Nt]\right) & X_{\mathrm{sum}} > \mathrm{P}_{\max}, \\ \mathbf{x}_{mi}^{(T_{\mathrm{NN}})}[1:Nt] + 1j \times \mathbf{x}_{mi}^{(T_{\mathrm{NN}})}[Nt+1:2Nt] & other, \end{cases} \tag{24}$$

of node features and edge features, respectively. The difference in the aggregation blocks for different UEs is reflected in the fact that the weights of their block are different, while the aggregation methods are the same, as shown in (22) and (23). By applying the element-wise mean function, the beam node obtains equal information from the other beam nodes. Besides, the features from the previous module are concatenated to retain the previous information.

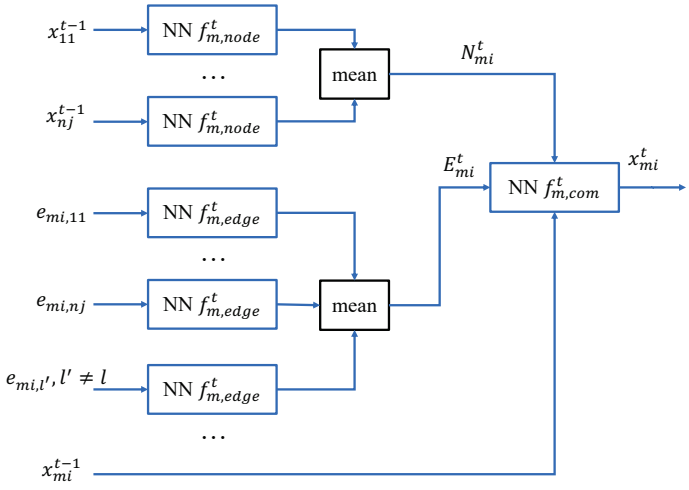


Fig. 6. The t-th updating module for mi-th node.

After aggregating the neighboring node features and neighboring edge features, the next part is the combining network, denoted as $\mathbf{x}_{mi}^{(t)} = f_{m,\text{com}}^{(t)}\left(\mathbf{x}_{mi}^{(t-1)}, \mathbf{N}_{mi}^{(t)}, \mathbf{E}_{mi}^{(t)} | \mathcal{W}_{m,3}^{(t)}\right)$. The difference in the combination blocks for different UEs is reflected in the fact that the weights of their block are different, while the combination methods are the same. After multiple node updating modules, the representation vectors of beam nodes contain sufficient information for designing the beamforming vectors.

3) **Normalization Module**: The last module is the normalization module. After T_{NN} node updating modules, the representation vectors are passed to a normalization layer to produce the beamforming matrix $\mathbf{w}_{mi}$, which ensuring the per AP power constraints on $\mathbf{w}_{mi}, m \in \mathcal{M}_l$. Representation vectors belonging to the i-th AP are normalized together, $\mathbf{w}_{mi} = f_{\text{norm}}\left(\mathbf{x}_{1i}^{(T_{\text{NN}})}, \quad \ldots, \mathbf{x}_{mi}^{(T_{\text{NN}})}, \ldots\right), \forall m \in \mathcal{M}_l$. Specifically, $\mathbf{w}_{mi}$ is computed as (24), where $X_{\text{sum}} = \sum_{m \in \mathcal{M}_l} \left\| \mathbf{x}_{mi}^{(T_{\text{NN}})} \right\|_2^2$.

The above is the entire MHSB-GNN structure. This structure has the following advantages. First, GNN increases the interpretability of the network because the modeling of the graph considers the priori information of the communication system, e.g., interference relations. Second, GNNs can generalize so that when the number of users changes, there is no need to retrain the network, and the trained network remains usable. Because when the number of users becomes small, the trained network invokes the relevant sub-modules to compute the representation vectors based on the number of users. We will demonstrate this advantage in the later experiments as well. The third advantage, also the core innovation of this network structure, is to design different node updating modules for beam nodes of different users. The network not only mines the unique features of the beams belonging to different users but also mines the similar features of the beams belonging to the same user. We introduce a comparison network in the later simulations, SGNN, with the same updating modules for all nodes. Please refer to the simulation section for the specific settings. The experiments demonstrate that the MHSB-GNN with different updating modules performs better than the SGNN with the same updating module regarding the same FLOPs and approximate running time.

C. Network Training

During the whole training process, the proposed MHSB-GNN is trained to adjust its parameters to maximize the lower bound of WSR by unsupervised learning with an Adam optimizer. The training samples and testing samples are generated by different channel realizations. The loss function is chosen to align with the objective function in (P2), denoted as

$$f_{\text{loss}} = -\frac{1}{B}\sum_{b=1}^{B}\sum_{m \in \mathcal{M}_l} \delta_m \mathring{r}_m, \tag{25}$$

where B is the batch size.

The proposed MHSB-GNN-based algorithm for cross-layer APS&BF design is outlined in Algorithm 3. The overall end-to-end training allows the network to design the beamforming matrix with the local instantaneous CSI and statistical interference CSI. Noting that the training phase is completed offline, the computational complexity is less of a concern.

Algorithm 3 MHSB-GNN algorithm for cross-layer APS&BF design

1: At the beginning of the τ-th frame, FP and AO-based algorithms are used to design the active AP set $\mathcal{S}$ and the beamforming vectors of the $\tau T_s + 1$-th time slot;
2: During the following time slots τ_s, beamforming vectors are computed according to the pre-trained MHSB-GNN.

D. Theoretical Proof

The machine learning-based alg is responsible for beamforming in the short term. APS is operated at the beginning of each frame.

Theorem 1: For any algorithm in distributed message passing (DMP), there exists a GNN that solves it.

Proof: This result is proved by following Theorem 2 in [33]. ■

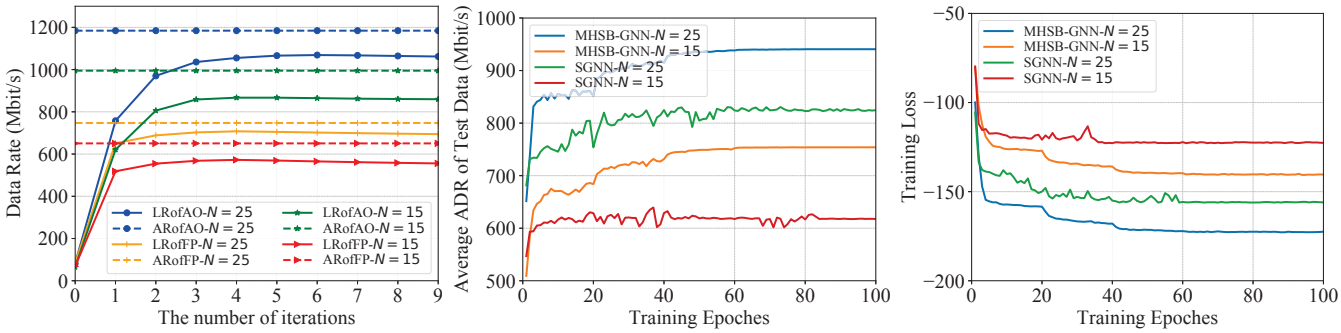
Theorem 2: FP-based algorithm is a DMP algorithm.

Proof: Please refer to Appendix B for proof. ■

Corollary 1: There exists a GNN that can solve problem (P2) without AP selecting variable $\mathcal{S}$.

V. NUMERICAL RESULTS

In this section, we provide performance evaluation for the proposed algorithms in comparison with the benchmark methods.



(a) Actual ADR of FP-based algorithms. (b) Average ADR of MHSB-GNN. (c) Training loss of MHSB-GNN.

Fig. 7. Coverage plot of proposed algorithms.

TABLE I
SIMULATION SETTINGS

Description	Value	Description	Value
Simulation Area	1.5km×1.5km	Bandwidth	10MHz
AP/UE distribution	unif. rand.	AP PSD	-47dBm/Hz
AP/UE antenna height	12/2m	Noise PSD	-169dBm/Hz
Carrier frequency	2GHz	σ_{sd}	8dB
Coherence bandwidth	200kHz	N_t	4

TABLE II
ARCHITECTURE OF FULLY CONNECTED NETWORK

Name	Size	Active function
f_{init}	$8 \times 128 \times 256$	ReLU
f'_{init}	$32 \times 128 \times 256$	ReLU
$f_{m,\text{node}}$	$256 \times 256 \times 512 \times 256$	ReLU
$f_{m,\text{edge}}$	$256 \times 256 \times 512 \times 256$	ReLU
f_{com}	$768 \times 512 \times 256 \times 256$	ReLU
f_{norm}	256×8	–

A. Simulation Setup

1) **CFN Setting**: Consider $L = 5$ groups and $K/L = 5$ APs, $M/L \in \{1, 2, 3, 4\}$ UEs in each group. The group-to-group distance is 800m. APs and UEs are randomly placed in each group with a minimum distance 50m from each other. The maximum transmit power spectrum density (PSD) for APs is -47dBm/Hz, and the background noise PSD is -169dBm/Hz over 10 MHz bandwidth [44]. The channel model is set as [44]: 1) Large-scale fading coefficients include path loss $\text{PL} = -30.5 - 36.7 \log_{10}(d_{mi})$ and uncorrelated shadow fading with standard deviation σ_{sd}; 2) Rayleigh channel fading with zero mean and unit variance. The simulation settings are reported in Tab. I.

2) **FP/AO:** APS&BF is designed by FP-based and AO-based algorithms with local CSI.

3) **MHSB-GNN:** Active AP set is determined by FP-based algorithm, and then BF is designed through Algorithm 3 with local CSI. MHSB-GNN is a 4-module network, including one initial module, $T_{\text{NN}} = 2$ node updating modules, and one normalization module. The sub-networks of MHSB-GNN are summarized in Tab. II. The MHSB-GNN is trained for 100 epochs, and the optimizer is Adam, with the initial learning rate set to 1, decayed by 10 every 20 epochs. The batch size B is set to 100. After randomly generating the locations of APs and UEs, we generate T_c times local channels $\{\mathbf{h}_{mi}^1, \ldots, \mathbf{h}_{mi}^{T_c}\}, m \in \mathcal{M}_l, i \in \mathcal{I}_l$ and interfering channels $\{\mathbf{h}_{mi}^1, \ldots, \mathbf{h}_{mi}^{T_c}\}, i \in \mathcal{I}_l^{'}$. We treat the T_c-th channel $\mathbf{h}_{mi}^{T_c}$ as the local instantaneous channel. The statistical interference CSI, channel covariance matrix, is calculated by the past $T_c - 1$ channels $\{\mathbf{h}_{mj}^1, \ldots, \mathbf{h}_{mj}^{T_c}\}$. As the system has $L = 5$ groups, the above process can generate L samples simultaneously. Repeat the above process for 1200 times. There are 6000 samples finally, including 5000 training and 1000 testing samples.

4) **SGNN**: SGNN is directly derived from the FP-based alg. The specific SGNN structure refers to the DMP derived from the FP-based alg. In this network, all nodes share the same node update module. SGNN has the same number of modules as MHSB-GNN. The difference between MHSB-GNN and SGNN is that all nodes in SGNN share the same updating module $f_{m,\text{node}} = f_{\text{node}}$, $f_{m,\text{edge}} = f_{\text{edge}}$, $f_{m,\text{com}} = f_{\text{com}}$. The SGNN is trained for 100 epochs, the optimizer is Adam, and the learning rate is set to 0.001.

5) **ESAO**: When N is determined, the active AP set is determined by the exhaustive search. Then, the AO-based alg with the determined variable $\mathcal{S}$ is performed under every selection.

B. Convergence Behavior

Fig. 7(a), Fig. 7(b), and Fig. 7(c) demonstrate the convergence performance of the proposed FP, AO, and MHSB-GNN APS&BF algorithms, providing evidence for the accuracy of these methods. The algorithms for simulations are represented by LRofFP, ARofFP, IRofOA, ARofOA, and MHSB-GNN. "LR" and "AR" mean the lower bound of the data rate and the actual data rate, respectively. For example, LRofFP means the lower rate of the FP-based algorithm. In Fig. 7(a), the

TABLE III
RUNNING TIME (S) WITH DIFFERENT NUMBERS OF UE AND AP

	$K=25$			$K=20$			$K=15$		
Alg.	$M=10$	$M=15$	$M=20$	$M=10$	$M=15$	$M=20$	$M=10$	$M=15$	$M=20$
AO-based	13.84	16.1240	26.78	11.97	14.48	14.17	10.68	11.14	12.25
FP-based	0.062	0.1654	0.285	0.0578	0.0684	0.1036	0.0534	0.0675	0.0910
MHSB-GNN	0.0598	0.1246	0.23825	0.0339	0.1067	0.1825	0.0219	0.0498	0.0917
SGNN	0.04686	0.156	0.2342	0.03485	0.0877	0.1595	0.01994	0.0498	0.0907

TABLE IV
FLOPS WITH DIFFERENT NUMBERS OF UE AND AP

	$K=25$			$K=20$			$K=15$		
Alg.	$M=10$	$M=15$	$M=20$	$M=10$	$M=15$	$M=20$	$M=10$	$M=15$	$M=20$
MHSB-GNN	0.3×10^5	0.5×10^5	0.7×10^5	0.29×10^5	0.43×10^5	0.57×10^5	0.22×10^5	0.32×10^5	0.43×10^5
SGNN	0.3×10^5	0.5×10^5	0.7×10^5	0.29×10^5	0.43×10^5	0.57×10^5	0.22×10^5	0.32×10^5	0.43×10^5

changing curves represent the convergence process of LR, while the invariant curves represent the actual achievable data rate (ADR). It can be seen that the lower bound of WSR monotonically increases and converges rapidly with the number of iterations, and all curves generally converge within seven iterations. It also can be observed that AO outperforms FP in sum-rate performance. As expected, the actual ADR of FP and AO are always higher than the lower bound of the data rate, providing further evidence for the accuracy of Lemma 1. Fig. 7(a) also illustrates the impact of the number of active APs on performance, and algorithms with a larger number of active APs can converge to a larger WSR due to higher spatial degrees of freedom. However, it is shown that 66% increase in transmit power results in 20% gain in ADR performance.

Fig. 7(c) plots the loss evaluated on the training set against the training epochs in a single group with $M/L = 3$ and $N/L = \{3, 4\}$. Fig. 7(b) plots the average ADR evaluated on the validation set. Both MHSB-GNN and SGNN converge within 60 epochs. The loss and test performance curves of MHSB-GNN exhibit a step-like character due to the changing learning rate with epochs. Changing the learning rate does not work in SGNN, and the curves of SGNN do not have a step-like character. It shows that MHSB-GNN converges in fewer iterations than SGNN, e.g., the former achieves a sum rate of 750Mbps within 10 epochs, but the latter needs 60 epochs, as shown in Fig. 7(b). Therefore, the training time for MHSB-GNN is significantly less than that of SGNN.

C. FLOPs and Running Time

The performance of FP-based alg is lower than that of AO-based alg with a finite number of iterations, as shown in Fig. 7(a). On the same device, the running time of AO-based alg is almost 100 times that of FP-based alg, as shown in Tab. III. AO-based alg contains two layers of looping algorithm. The number of iterations for the outer algorithm is the same as that for FP-based alg, while the number of iterations for the inner algorithm is unknown. Usually, the interior-point method implements the internal algorithm through the CVX toolbox on Matlab. The interior-point method goes through many iterations to satisfy a specific error, which results in the long running time and high complexity.

As shown in Tab. III, SGNN, MHSB-GNN, and FP-based alg have the same order of magnitude of running time. The running time of these three algorithms is less affected by the different numbers of users. In contrast, the running time of AO-based is highly influenced by the different number of users. The FLOPs of SGNN and MHSB-GNN also show linear growth with the changing number of users. Moreover, from the previous complexity derivation, it is clear that the complexity of AO-based shows exponential growth with the number of users. From the experimental results Fig. 7, Tab.III and Tab. IV, SGNN and MHSB-GNN have similar FLOPs and the same order of magnitude of running time. However, MHSB-GNN outperforms SGNN by 13.25% when $N = 25$.

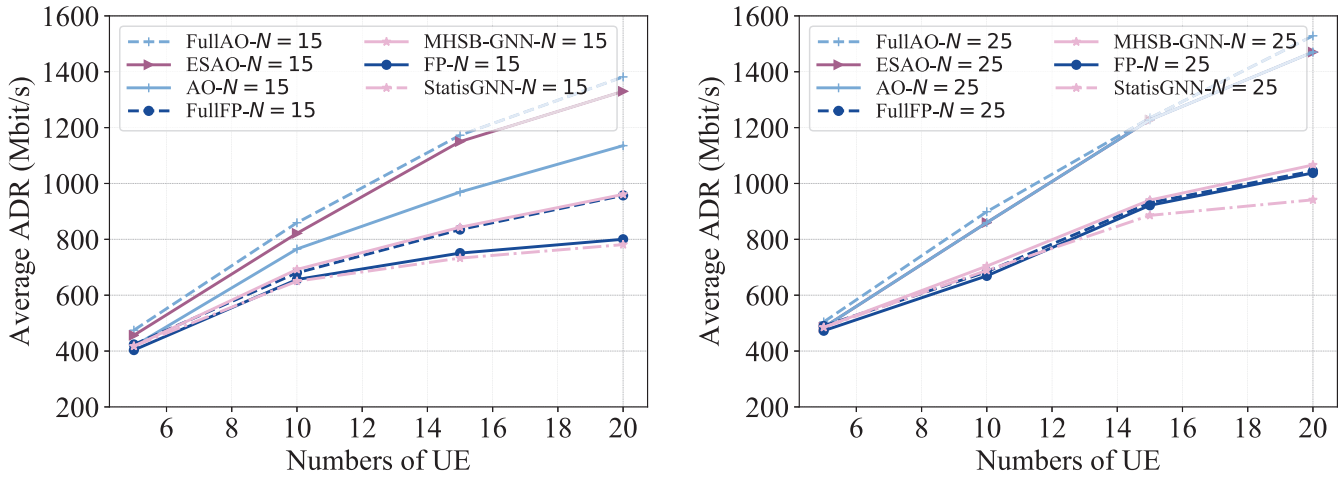
TABLE V
THE GENERALIZATION OF MHSB-GNN TO DIFFERENT NUMBER OF UES. (UNIT: MBIT/S)

Alg.	$M_1=2$	$M_1=3$	$M_1=4$
MHSB-GNN	131.488	162.145	210.140
MHSB-GNN-Re-train	145.430	188.246	210.140
SGNN	130.516	161.310	185.213
SGNN-Re-train	137.488	162.141	185.213

D. The Generalization of MHSB-GNN

Tab. V shows the generalization performance of MHSB-GNN at different numbers of UEs. There are still $L = 5$ groups

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2024.3361900



(a) $N = 15$ (b) $N = 25$

Fig. 8. The average ADR achieved by different algorithms under different numbers of users and active APs.

and there are $|\mathcal{M}_{l'}| = 4$ UEs in $l^{'}, l^{'} \neq l$-th group. The number of UEs in l-th is set to $|\mathcal{M}_l| = \{2, 3, 4\}$. The networks MHSB-GNN and SGNN are first trained with $|\mathcal{M}_l| = 4, \forall l$, and then tested under the different numbers of UEs without retraining the network. We also compare with the retrained MHSB-GNN and SGNN, represented by "MHSB-GNN-Re-train" and "SGNN-Re-train", independently retrained and tested under different UEs. The average ADR in l-th group is obtained by averaging over all 100 test channel realizations. It is observed that MHSB-GNN consistently outperforms SGNN under the different numbers of UEs regardless of whether the testing network is retrained or not. SGNN performs better than MHSB-GNN in terms of generalization. Specifically, the gaps between MHSB-GNN and MHSB-GNN-Re-train are larger than the gaps between SGNN and SGNN-Re-train. When $|\mathcal{M}_1| = 3$, the achievable ADR of SGNN is almost 0.5% lower that of SGNN-Re-train. The fundamental reason is that MHSB-GNN designs the specific node updating modules for different UEs, while different UEs in SGNN share the same updating module.

E. Performance Versus the Number of UEs

Fig. 8 shows the average ADR evaluated on 100 channels at the different UE numbers with $N = \{15, 25\}$. "FullFP" and "FullAO" mean that the FP-based and AO-based algorithms are applied with global instantaneous CSI. "StatisGNN" means that the local covariance information $\mathbf{R}_{mi} = \mathbb{E}\left\{\mathbf{h}_{mi}^{\mathrm{H}}\mathbf{h}_{mi}\right\} \in \mathbb{C}^{N_t \times N_t}$ instead of local instantaneous CSI $\mathbf{h}_{mi}$ is used as the input of the MHSB-GNN. Fig. 8(a) shows the average ADR performance with APs ($N = 15$) of different algorithms. It is observed that MHSB-GNN performs better than FP-based alg. Especially when $M = 20$, the average ADR achieved by the MHSB-GNN is 18.8% higher than FP-based alg. Note that the computational time of MHSB-GNN is in the same order as that of FP-based alg from Tab. III, which is acceptable in a practical system. Fig. 8(b) illustrates the simulation results without APS ($N = 25$). It is observed that the gap between FP-based alg and MHSB-GNN becomes narrow, although the average ADR of the latter remains a little higher than that of the former. The main reason is that the number of APs becomes large, and thus, the MHSB-GNN is difficult to learn well. StatisGNN achieves lower performance than MHSB-GNN because the input of StatisGNN is not accurate enough, and is all statistical CSI.

It is obtained from Fig. 8(b) that the performance loss caused by applying local CSI instead of full CSI is small, less than 1%. Replacing full CSI with local CSI helps reduce the interaction overhead among groups and complexity, which is acceptable in a practical system. In Fig. 8(a), the gap between full CSI and local CSI increases in average ADR as the number of UEs increases. The reason is that when M becomes larger, the more estimation error accumulation in the optimization objective function. Obviously, the best average ADR is achieved by ESAO, followed by AO-based alg. However, this performance advantage comes at the cost of computational complexity and long running time. Therefore, MHSB-GNN is more practical than AO and FP-based algorithms as it balances the ADR performance and complexity.

F. Performance Versus the Transmitting Power

Fig. 9 investigates the impact of transmitting power on the performance of different APS&BF schemes with $N = 15$ and $N = 25$. It can be seen that the ADR of all APS&BF schemes increases with the increasing transmitting power. Meanwhile, as the transmitting power increases, the increase in power brings fewer and fewer performance gains. This is because the power of the useful signals and the interfering signals increases at the same time. When the number of active APs is $N = 15$, ESAO can obtain the best ADR performance with the highest complexity cost. Additionally, it is observed that the FP-based alg and MHSB-GNN alg benefit more from the increase of $\mathrm{P}_{\max}$ than other algorithms.

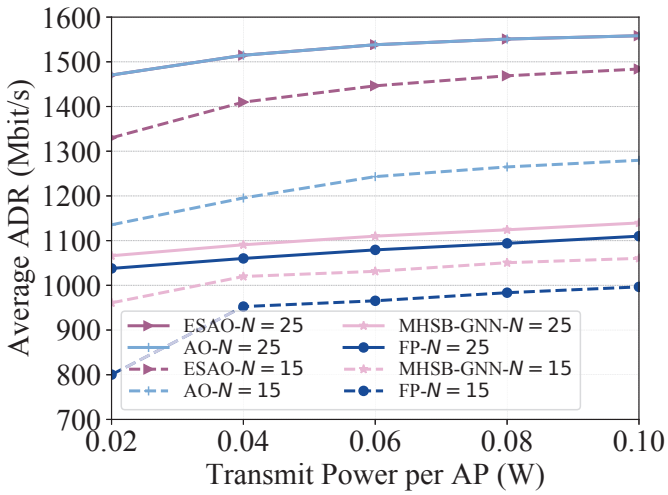


Fig. 9. The average sum rate of different algorithms with different transmitting power.

VI. CONCLUSION

This article investigated a distributed maximizing the downlink weighted sum rate (WSR) with local CSI in non-coherent CFN. A joint optimization scheme of APS&BF was formulated as a MINP problem, which was challenging. To solve this problem, we proposed three distributed algorithms, named FP, AO, and MHSB-GNN-based algorithms. The novel MHSB-GNN took local instantaneous CSI and statistical interference CSI as input. The MHSB-GNN benefited from the different node updating modules for different users, which introduced more prior information into the graph and mined specific features of different users. As a result, MHSB-GNN obtained better performance than the baseline, SGNN. Moreover, for the interpretability and theoretical guarantees of MHSB-GNN, the equivalence between GNN and FP-based algorithms is provided. Simulation results showed the superiority of the proposed MHSB-GNN over the FP-based algorithm in terms of data rate with comparable computations. This research validates the generalization of the trained GNN to a smaller number of users. In practice, one should also take generalizing the trained GNN to a larger number of users into account, which is left as future work.

APPENDIX A
PROOF OF THE LEMMA 1

We can easily find the sum rate function is convex, which can be checked by second order derivative of rate function is always larger than 0. By leveraging the Jensen's inequality, a lower bound of the average achievable date rate of user m is given by

$$\mathbb{E}\left\{\hat{r}_m\right\} \triangleq \log\left(1+\widetilde{\mathrm{SINR}}_m\right) \leq \mathbb{E}\left\{r_m\right\}, \tag{26}$$

where $\mathbb{E}\left\{r_m\right\}$ is the average achievable rate of UE m, $\widetilde{\mathrm{SINR}}_m = \mathcal{A}_m/\left(\mathcal{B}_{1,m}-\mathcal{A}_m+\overline{\mathcal{B}}_{2,m}+\sigma_m^2\right)$. $\overline{\mathcal{B}}_{2,m} = \sum_{j\in\mathcal{I}_{l'}} s_j \sum_{n\in\mathcal{M}_{l'}} \mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{R}_{mj}\mathbf{w}_{nj}^{\mathrm{H}}\right)$ is the expectation value of the inter-group interference $\mathcal{B}_{2,m}$ and $\mathbf{R}_{mj} = \mathbb{E}\left\{\mathbf{h}_{mj}^{\mathrm{H}}\mathbf{h}_{mj}\right\} \in \mathbb{C}^{N_t\times N_t}$ is the expected correlation matrix of the channel vector $\mathbf{h}_{mj}$. Then, an upper bound of average inter-cell interference $\widetilde{\mathcal{B}}_{2,m}$ is given by

$$\begin{aligned}\widetilde{\mathcal{B}}_{2,m} &\leq \mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{w}_{nj}^{H}\mathbf{R}_{mj}\right) \overset{a}{\leq} \mathrm{Tr}\left(\mathbf{w}_{nj}\mathbf{w}_{nj}^{H}\right) Tr\left(\mathbf{R}_{mj}\right) \\ &\overset{b}{\leq} P_j \mathrm{Tr}\left(\mathbf{R}_{mj}\right) \triangleq \overline{\mathcal{B}}_{2,m},\end{aligned} \tag{27}$$

where (27-a) follows from the $\mathrm{Tr}\left(AB\right) \leq \mathrm{Tr}\left(A\right)\mathrm{Tr}\left(B\right)$; (27-b) follows from the power constraint. Replacing the inter-group interference $\mathcal{B}_{2,m}$ with the inter-group interference upper bound $\overline{\mathcal{B}}_{2,m}$, this completes the proof.

APPENDIX B
PROOF OF THE THEOREM 2

There are two kinds of nodes in the graph, beam nodes $V_1 = \{v_{mi} | m \in \mathcal{M}_l, i \in \mathcal{I}_l\}$ and neighboring group nodes $V_2 = \{v_{l'} | l^{'} \neq l\}$. The node feature of v_{mi} is $[\delta_{mi}, \sigma_{mi}, \mathbf{h}_{mi}]$. The internal state of node v_{mi} at the $(t-1)$-th iteration is $[\gamma_{mi}^{(t-1)}, \mathbf{y}_{mi}^{(t-1)}, \mathbf{w}_{mi}^{(o-1)}, \delta_{mi}, \sigma_{mi}, \mathbf{h}_{mi}]$. If $n \neq m$, there is an intra-group interference edge between the the node v_{mi} to the node v_{nj}. If $n = m$ and $j \neq i$, the beam node belongs to the same user m. The edge feature of the edge (nj, mi) is $\mathbf{e}_{nj,mi} = [\mathbf{h}_{mj}, \mathbf{h}_{ni}]$. The node feature of $v_{l'}$ is $\mathrm{P}_{\max}$. An edge is drawn from the node $v_{l'}$ to the node v_{mi} is the inter-group interference edge. The edge feature of the edge $(l^{'}, mi)$ is $[\ldots, \mathbf{R}_{mj}, , \ldots], j \in \mathcal{I}_{l'}$.

Assume the FP-based alg need T_{FP} iterations, which means the corresponding multi-set broadcasting distributed local algorithms (MB-DLA) has at least $2T_{\mathrm{FP}}$ iterations. In the corresponding MB-DLA, the variables γ_{mi} and $\mathbf{y}_{mi}$ are updated at the odd iterations while $\mathbf{w}_{mi}$ is updated at the even iterations. Specifically, at the t-th iteration with t being an odd number, the (n,j)-th node broadcasts its state $\mathbf{w}_{nj}$ along its edges. The edge $(nj, mi), n \neq m$ processes the first kind of message by forming $m_{nj,mi}^{(1)} = \mathbf{h}_{mj}\mathbf{w}_{nj}^{(t-1)\mathrm{H}}\mathbf{w}_{nj}^{(t-1)}\mathbf{h}_{mj}^{\mathrm{H}}$ and the node (m,i) receives the message set $\{m_{nj,mi}^{(1)}\}$. The agent (m,i) sums over the messages $M_{mi}^{(1)} = \sum_{j\in\mathcal{I}_l}\sum_{n\neq m} m_{nj,mi}^{(1)}$. The edge $(l^{'}, mi)$ processes the second kind of message by forming $m_{l',mi}^{(2)} = \mathrm{P}_{\max}\sum_{j\in\mathcal{I}_{l'}} \mathrm{Tr}(R_{mj'})$ and the node (m,i) receives the message set $\{m_{l'j',mi}^{(2)}\}$. The agent (m,i) sums over the messages $M_{mi}^{(2)} = \sum_{l'\neq l} m_{l'j',mi}^{(2)}$. The edge $(nj, mi), n = m$ processes the third kind of message by forming $m_{mj,mi}^{(3)} = \mathbf{h}_{mj}\mathbf{w}_{mj}^{(t-1)\mathrm{H}}\mathbf{w}_{mj}^{(t-1)}\mathbf{h}_{mj}^{\mathrm{H}}$ and the node (m,i) receives the message set $\{m_{mj,mi}^{(3)}\}$. The agent (m,i) sums over the messages $M_{mi}^{(3)} = \sum_{j\in\mathcal{I}_l}\sum_{n=m} m_{mj,mi}^{(3)}$. Then, the (m,i)-th node updates its internal state as $\gamma_{m,i}^{(t)} = \frac{\left|\mathbf{h}_{mi}\mathbf{w}^{(t-1)\mathrm{H}}\right|^2}{M_{mi}^{(1)}+M_{mi}^{(2)}+\sigma_{mi}}$ and $\mathbf{y}_{mi} = (M_{mi}^{(1)} + M_{mi}^{(2)} + M_{mi}^{(3)} + \sigma_{mi})^{-1}\sqrt{\delta_m(1+\gamma_m)}\mathbf{h}_{mi}\mathbf{w}_{mi}^{(t-1)\mathrm{H}}$. Specifically, we construct the following process. When $n \neq m$,

$$\begin{aligned}&\mathbf{x}_{nj}^{(t-1)} = [\gamma_{nj}^{(t-1)}, \mathbf{y}_{nj}^{(t-1)}, \mathbf{w}_{nj}^{(t-1)}, \delta_{nj}, \sigma_{nj}, \mathbf{h}_{nj}], \\ &\mathbf{e}_{nj,mi} = [\mathbf{h}_{mj}, \mathbf{h}_{ni}], \\ &\mathbf{y}_{mi} = h_{1,0}^{(t)}(\mathbf{x}_{mi}^{(t-1)}) = \mathbf{x}_{mi}^{(t-1)}, \\ &m_{nj,mi}^{(1)} = \mathbf{h}_{mj}\mathbf{w}_{nj}^{(t-1)\mathrm{H}}\mathbf{w}_{nj}^{(t-1)}\mathbf{h}_{mj}^{\mathrm{H}} = h_{2,0}^{(t)}(\mathbf{y}_{mi}, \mathbf{e}_{nj,mi}),\end{aligned}$$

$$M_{mi}^{(1)} = g_{1,0}^{(t)}(\{m_{nj,mi}^{(1)}\}) = \sum_{j\in\mathcal{I}_l}\sum_{n\neq m} m_{nj,mi}^{(1)}.$$

When $n = m$,

$$\begin{aligned}
&\mathbf{x}_{mj}^{'(t-1)} = [\gamma_{mj}^{(t-1)}, \mathbf{y}_{mj}^{(t-1)}, \mathbf{w}_{mj}^{(t-1)}, \delta_{mj}, \sigma_{mj}, \mathbf{h}_{mj}],\\
&\mathbf{e}_{mj,mi} = [\mathbf{h}_{mj}, \mathbf{h}_{mi}],\\
&\mathbf{y}_{mi}^{'} = h_{1,1}^{(t)}(\mathbf{x}_{mi}^{'(t-1)}) = \mathbf{x}_{mi}^{'(t-1)},\\
&m_{mj,mi}^{(3)} = \mathbf{h}_{mj}\mathbf{w}_{mj}^{(t-1)\mathrm{H}}\mathbf{w}_{mj}^{(t-1)}\mathbf{h}_{mj}^{\mathrm{H}} = h_{2,1}^{(t)}(\mathbf{y}_{mi}^{'}, \mathbf{e}_{mj,mi}),\\
&M_{mi}^{(3)} = g_{1,1}^{(t)}(\{m_{mj,mi}^{(3)}\}) = \textstyle\sum_{j\in\mathcal{I}_l} m_{mj,mi}^{(3)},\\
&\mathbf{x}_{l'j'} = [\mathrm{P}_{\max}],\\
&\mathbf{e}_{l',mi} = [\ldots, \mathbf{R}_{mj}, \ldots],\\
&\mathbf{y}_{mi}^{(2)} = h_{1,2}^{(t)}(\mathbf{x}_{mi}^{'(t-1)}) = \mathbf{x}_{mi}^{'(t-1)},\\
&m_{l',mi}^{(2)} = \textstyle\sum_{j\in\mathcal{I}_{l'}} \mathrm{Tr}(\mathbf{R}_{mj})\mathrm{P}_{\max} = h_{2,2}^{(t)}(\mathbf{y}_{mi}^{(2)}, \mathbf{e}_{l',mi}),\\
&M_{mi}^{(2)} = g_{1,2}^{(t)}(\{m_{l',mi}^{(2)}\}) = \textstyle\sum_{l'\neq l} m_{l',mi}^{(2)},\\
&\gamma_{mi}^{(t)} = (M_{mi}^{(1)} + M_{mi}^{(2)} + \delta_{mi}^2)^{-1}\left|\mathbf{h}_{mi}\mathbf{w}_{mi}^{(t-1)\mathrm{H}}\right|^2,\\
&\mathbf{y}_{mi} = (\sigma_{mi}^2 + M_{mi}^{(1)} + M_{mi}^{(2)} + M_{mi}^{(3)})^{-1}\\
&\qquad\sqrt{\delta_{mi}(1+\textstyle\sum_{i\in\mathcal{I}_l}\gamma_{mi}^{(t-1)})}\mathbf{h}_{mi}\mathbf{w}_{mi}^{(t-1)\mathrm{H}},\\
&\mathbf{x}_{mi}^{(t)} = g_2^{(t)}(\mathbf{x}_{mi}^{(t-1)}, M_{mi}^{(1)}, M_{mi}^{(2)}, M_{mi}^{(3)})\\
&\qquad = [\gamma_{mi}^{(t-1)}, \mathbf{y}_{mi}^{(t-1)}, \mathbf{w}_{mi}^{(t-1)}, \delta_{mi}, \sigma_{mi}, \mathbf{h}_{mi}].
\end{aligned}$$

At the t-th iteration where t is even, the (n,j)-th node broadcasts its state $[\gamma_{nj}^{(t-1)}, \mathbf{y}_{nj}^{(t-1)}]$ along its edges. The edge (nj, mi) processes the first kind of message by forming $m_{nj,mi} = \mathbf{h}_{mi}^{\mathrm{H}}\mathbf{y}_{nj}\mathbf{y}_{nj}^{\mathrm{H}}\mathbf{h}_{mi}$. Node (m,i) receives the message set $\{m_{nj,mi}^{(1)}, n \neq m, j \in \mathcal{I}_l\}$. The agent (m,i) sums over the messages $M_{mi}^{(1)} = \sum_{j\in\mathcal{I}_l}\sum_{n\neq m} m_{nj,mi}^{(1)}$. The edge $(l^{'}, mi)$ processes the second kind of message by forming $m_{l',mi}^{(2)} = \mathrm{P}_{\max}\sum_{j\in\mathcal{I}_{l'}}\mathrm{Tr}(R_{mj'})$ and the node (m,i) receives the message set $\{m_{l'j,mi}^{(2)}\}$. The agent (m,i) sums over the messages $M_{mi}^{(2)} = \sum_{l'\neq l} m_{l'j',mi}^{(2)}$. The edge (nj, mi) processes the third kind of message by forming $m_{mj,mi}^{(3)} = \mathbf{h}_{mi}^{\mathrm{H}}\mathbf{y}_{mj}\mathbf{y}_{mj}^{\mathrm{H}}\mathbf{h}_{mi}$ and the node (m,i) receives the message set $\{m_{mj,mi}^{(3)}, n = m, j \in \mathcal{I}_l\}$. The agent (m,i) sums over the messages $M_{mi}^{(3)} = \sum_{j\in\mathcal{I}_l}\sum_{n=m} m_{mj,mi}^{(3)}$. Then the (m,i)-th node updates its internal state as $\mathbf{w}_{mi}^{(t)} = (M_{mi}^{(1)} + M_{mi}^{(2)} + M_{mi}^{(3)} + \eta_{mi}^{*}\mathbf{I})^{-1}\sqrt{\delta_{mi}(1+\sum_{i\in\mathcal{I}_l}\gamma_{mi}^{(t-1)})}\mathbf{h}_{mi}\mathbf{y}_{mi}^{(t-1)\mathrm{H}}$. Specifically, we construct the following process. When $n \neq m$,

$$\begin{aligned}
&\mathbf{x}_{nj}^{(t-1)} = [\gamma_{nj}^{(t-1)}, \mathbf{y}_{nj}^{(t-1)}, \mathbf{w}_{nj}^{(t-1)}, \delta_{nj}, \sigma_{nj}, \mathbf{h}_{nj}],\\
&\mathbf{e}_{nj,mi} = [\mathbf{h}_{mj}, \mathbf{h}_{ni}],\\
&\mathbf{y}_{mi} = h_{1,0}^{(t-1)}(\mathbf{x}_{mi}^{(t-1)}) = \mathbf{x}_{mi}^{(t-1)},\\
&m_{nj,mi}^{(1)} = \mathbf{h}_{mj}\mathbf{y}_{nj}^{(t-1)\mathrm{H}}\mathbf{y}_{nj}^{(t-1)}\mathbf{h}_{mj}^{\mathrm{H}} = h_{2,0}^{(t)}(\mathbf{y}_{mi}, \mathbf{e}_{nj,mi}),\\
&M_{mi}^{(1)} = g_{1,0}^{(t)}(\{m_{nj,mi}^{(1)}\}) = \textstyle\sum_{j\in\mathcal{I}_l}\sum_{n\neq m} m_{nj,mi}^{(1)}.
\end{aligned}$$

When $n = m$,

$$\begin{aligned}
&\mathbf{x}_{mj}^{'(t-1)} = [\gamma_{mj}^{(t-1)}, \mathbf{y}_{mj}^{(t-1)}, \mathbf{w}_{mj}^{(t-1)}, \delta_{mj}, \sigma_{mj}, \mathbf{h}_{mj}],\\
&\mathbf{e}_{mj,mi} = [\mathbf{h}_{mj}, \mathbf{h}_{mi}],\\
&\mathbf{y}_{mi}^{'} = h_{1,1}^{(t)}(\mathbf{x}_{mi}^{'(t-1)}) = \mathbf{x}_{mi}^{'(t-1)},\\
&m_{mj,mi}^{(3)} = \mathbf{h}_{mj}\mathbf{y}_{mj}^{(t-1)\mathrm{H}}\mathbf{y}_{mj}^{(t-1)}\mathbf{h}_{mj}^{\mathrm{H}} = h_{2,1}^{(t)}(\mathbf{y}_{mi}^{'}, \mathbf{e}_{mj,mi}),\\
&M_{mi}^{(3)} = g_{1,1}^{(t)}(\{m_{mj,mi}^{(3)}\}) = \textstyle\sum_{j\in\mathcal{I}_l} m_{mj,mi}^{(3)},\\
&\mathbf{x}_{l'} = [\mathrm{P}_{\max}],\\
&\mathbf{e}_{l'j',mi} = [\ldots, \mathbf{R}_{mj}, \ldots],\\
&\mathbf{y}_{mi}^{(2)} = h_{1,2}^{(t)}(\mathbf{x}_{mi}^{'(t-1)}) = \mathbf{x}_{mi}^{'(t-1)},\\
&m_{l',mi}^{(2)} = \textstyle\sum_{j\in\mathcal{I}_{l'}} \mathrm{Tr}(\mathbf{R}_{mj})\mathrm{P}_{\max} = h_{2,2}^{(t)}(\mathbf{y}_{mi}^{(2)}, \mathbf{e}_{l'j',mi}),\\
&M_{mi}^{(2)} = g_{1,2}^{(t)}(\{m_{l'j',mi}^{(2)}\}) = \textstyle\sum_{l'\neq l} m_{l',mi}^{(2)},\\
&\mathbf{w}_{mi} = (\eta_{mi}\mathbf{I} + M_{mi}^{(1)} + M_{mi}^{(2)} + M_{mi}^{(3)})^{-1}\\
&\qquad\sqrt{\delta_{mi}(1+\textstyle\sum_{i\in\mathcal{I}_l}\gamma_{mi}^{(t-1)})}\mathbf{h}_{mi}\mathbf{y}_{mi}^{(t-1)\mathrm{H}},\\
&\eta_{mi}^{*} = \arg\min_{\eta_{mi}\geq 0, \mathrm{Tr}\left(\mathbf{w}_{mi}\mathbf{w}_{mi}^{\mathrm{H}}\right)\leq P_i} \eta_{mi},\\
&\mathbf{w}_{mi} = (\eta_{mi}^{*}\mathbf{I} + M_{mi}^{(1)} + M_{mi}^{(2)} + M_{mi}^{(3)})^{-1}\\
&\qquad\sqrt{\delta_{mi}(1+\textstyle\sum_{i\in\mathcal{I}_l}\gamma_{mi}^{(t-1)})}\mathbf{h}_{mi}\mathbf{y}_{mi}^{(t-1)\mathrm{H}},\\
&\mathbf{x}_{mi}^{(t)} = g_2^{(t)}(\mathbf{x}_{mi}^{(t-1)}, M_{mi}^{(1)}, M_{mi}^{(2)}, M_{mi}^{(3)})\\
&\qquad = [\gamma_{mi}^{(t-1)}, \mathbf{y}_{mi}^{(t-1)}, \mathbf{w}_{mi}^{(t-1)}, \delta_{mi}, \sigma_{mi}, \mathbf{h}_{mi}].
\end{aligned}$$

This completes the proof of the FP-based algorithm (without selection matrix $\mathcal{S}$) being a DMP.

REFERENCES

[1] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang, J. Wang *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, no. 1, pp. 1–74, 2021.

[2] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.

[3] V. Ranjbar, A. Girycki, M. A. Rahman, S. Pollin, M. Moonen, and E. Vinogradov, "Cell-free mMIMO support in the O-RAN architecture: A phy layer perspective for 5G and beyond networks," *IEEE Commun. Stand. Mag.*, vol. 6, no. 1, pp. 28–34, 2022.

[4] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 957–975, 2020.

[5] Ö. T. Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *arXiv preprint arXiv:2108.02541*, 2021.

[6] S. Buzzi, C. D'Andrea, A. Zappone, and C. D'Elia, "User-centric 5G cellular networks: Resource allocation and comparison with the cell-free massive MIMO approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1250–1264, 2019.

[7] V. Palhares, A. R. Flores, and R. C. de Lamare, "Iterative access point selection, MMSE precoding and power allocation for cell-free networks," *arXiv preprint arXiv:2104.05165*, 2021.

[8] I. L. Shakya and F. H. Ali, "Joint access point selection and interference cancellation for cell-free massive MIMO," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1313–1317, 2020.

[9] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. Srinivas, "Resource allocation and scheduling in non-coherent user-centric cell-free MIMO," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[10] S. He, Z. An, J. Zhu, M. Zhang, Y. Huang, and Y. Zhang, "Cross-layer optimization: Joint user scheduling and beamforming design with QoS support in joint transmission networks," *IEEE Trans. Commun.*, 2022.

[11] 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.*, 2022.

[12] M. Hong, R. Sun, H. Baligh, and Z.-Q. Luo, "Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 226–240, 2013.

[13] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, 2016.

[14] Y. Shi, J. Zhang, and K. B. Letaief, "Group sparse beamforming for green cloud-RAN," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2809–2823, 2014.

[15] J. Clausen, "Branch and bound algorithms-principles and examples," *Department of Computer Science, University of Copenhagen*, pp. 1–30, 1999.

[16] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," in *50 Years of Integer Programming 1958-2008*. Springer, 2010, pp. 105–132.

[17] S. Boyd and J. Mattingley, "Branch and bound methods," *Notes for EE364b, Stanford University*, pp. 2006–07, 2007.

[18] 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, 2023.

[19] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted $l1$ minimization," *J.Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, 2008.

[20] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, 2011.

[21] A. A. Khan, R. S. Adve, and W. Yu, "Optimizing downlink resource allocation in multiuser MIMO networks via fractional programming and the hungarian algorithm," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5162–5175, 2020.

[22] E. Chen, M. Tao, and Y.-F. Liu, "Joint base station clustering and beamforming for non-orthogonal multicast and unicast transmission with backhaul constraints," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6265–6279, 2018.

[23] K. Shen and W. Yu, "Fractional programming for communication systems—part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, 2018.

[24] ——, "Fractional programming for communication systems—part II: Uplink scheduling via matching," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2631–2644, 2018.

[25] M. Kulin, T. Kazaz, E. De Poorter, and I. Moerman, "A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer," *Electronics*, vol. 10, no. 3, p. 318, 2021.

[26] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2838–2849, 2020.

[27] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, 2019.

[28] L. Schynol and M. Pesavento, "Coordinated sum-rate maximization in multicell MU-MIMO with deep unrolling," *arXiv preprint arXiv:2202.10371*, 2022.

[29] J. Zhang, Y. Huang, J. Wang, X. You, and C. Masouros, "Intelligent interactive beam training for millimeter wave communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2034–2048, 2020.

[30] Z. Zhang, M. Tao, and Y.-F. Liu, "Learning to beamform in joint multicast and unicast transmission with imperfect CSI," *IEEE Trans. Commun.*, 2023.

[31] H. Hojatian, J. Nadal, J.-F. Frigon, and F. Leduc-Primeau, "Decentralized beamforming for cell-free massive MIMO with unsupervised learning," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1042–1046, 2022.

[32] R. Wang, W. Dai, and Y. Jiang, "Distributed learning for uplink cell-free massive MIMO networks," *IEEE Trans. Commun.*, 2023.

[33] Y. Shen, J. Zhang, S. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Trans. Wireless Commun.*, 2022.

[34] 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 Global Commun. Conf.*, Dec. 2021, pp. 01–06.

[35] 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. Wireless Commun.*, 2022.

[36] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "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, 2020.

[37] C. Chen, S. Xu, J. Zhang, and J. Zhang, "A distributed machine learning-based approach for IRS-enhanced cell-free MIMO networks," *arXiv preprint arXiv:2301.08077*, 2023.

[38] 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, 2023.

[39] X. Yan, Z. Wang, Y. Jia, Y. Huang, and L. Yang, "Cross-layer optimization of access point selection and beamforming in non-coherent cell free network," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*. Glasgow, United Kingdom,, 2023, pp. 1–6.

[40] C. Pan, H. Ren, M. Elkashlan, A. Nallanathan, and L. Hanzo, "The non-coherent ultra-dense C-RAN is capable of outperforming its coherent counterpart at a limited fronthaul capacity," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2549–2560, 2018.

[41] X. Chen, A. Liu, Y. Cai, V. K. Lau, and M.-J. Zhao, "Randomized two-timescale hybrid precoding for downlink multicell massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4152–4167, 2019.

[42] M. Hua and Q. Wu, "Joint dynamic passive beamforming and resource allocation for IRS-aided full-duplex WPCN," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 4829–4843, 2021.

[43] 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, 2021.

[44] Z. H. Shaik, E. Björnson, and E. G. Larsson, "Cell-free massive MIMO with radio stripes and sequential uplink processing," in *Proc. IEEE Int. Conf. Commun. Workshops*. IEEE, Jun. 2020, pp. 1–6.

Xuanhong Yan (Student Member, IEEE) received the B.Eng. degree in communication engineering from the College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin, China, in 2016, and the M.S. degree in Electronic and Communication Engineering from the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China, in 2018. She is currently pursuing a Ph.D. degree in information and communication engineering with the School of Information Science and Engineering, Southeast University, Nanjing, China. Her research interests mainly focus on intelligent wireless communications.

Zheng Wang (Senior Member, IEEE) received the B.S. degree in electronic and information engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009, and the M.S. degree in communications from University of Manchester, Manchester, U.K., in 2010. He received the Ph.D degree in communication engineering from Imperial College London, UK, in 2015.

Since 2021, he has been an Associate Professor in the School of Information and Engineering, Southeast University (SEU), Nanjing, China. From 2015 to 2016 he served as a Research Associate at Imperial College London, UK. From 2016 to 2017 he was an senior engineer with Radio Access Network R&D division, Huawei Technologies Co.. From 2017 to 2020 he was an Associate Professor at the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His current research interests include massive MIMO systems, machine learning and data analytics over wireless networks, and lattice theory for wireless communications.

Yi Jia (Student Member, IEEE) received the B.Eng. degree in information engineering from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2019, where she is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering. Her research interests mainly focus on intelligent wireless communications.

Zhengming Zhang received the B.S. degree in electronic information science and technology from Nanjing Agricultural University, Nanjing, China, in 2016. He received the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering, Southeast University. His current research interests include wireless big data, distributed machine learning, 5G mobile networks, UAV aided communication, and resource management.

Yongming Huang (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, China, in 2007.

Since March 2007, he has been a Faculty with the School of Information Science and Engineering, Southeast University, where he is currently a Full Professor. During 2008-2009, he visited the Signal Processing Lab, Royal Institute of Technology, Stockholm, Sweden. He has authored or coauthored more than 200 peer-reviewed papers, and holds more than 80 invention patents. His research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He submitted around 20 technical contributions to IEEE standards, and was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He was an Associate Editor for the IEEE Transactions on Signal Processing and a Guest Editor of the IEEE Journal Selected Areas in Communications. He is currently an Editor-at-Large of the IEEE Open Journal of the Communications Society and an Associate Editor for the IEEE Wireless Communications Letters.