IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS Connecting the Mobile World | IEEE Transactions on Machine Learning in Communications and Networking

Received 21 April 2023; revised 22 September 2023, 3 January 2024, and 10 March 2024; accepted 13 May 2024.
Date of publication 21 May 2024; date of current version 30 May 2024.

The associate editor coordinating the review of this article and approving its publication was A. Savard.

Digital Object Identifier 10.1109/TMLCN.2024.3403789

Optimal Access Point Centric Clustering for Cell-Free Massive MIMO Using Gaussian Mixture Model Clustering

PIALY BISWAS [1] (Member, IEEE), RANJAN K. MALLIK [2] (Fellow, IEEE), AND KHALED B. LETAIEF [3] (Fellow, IEEE)

[1] Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01062 Dresden, Germany
[2] Department of Electrical Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India
[3] Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong

CORRESPONDING AUTHOR: P. BISWAS (pialybiswas23@gmail.com)

The work of Ranjan K. Mallik was supported in part by the Science and Engineering Research Board, a Statutory Body of the Department of Science and Technology, Government of India, under the J. C. Bose Fellowship. The work of Khaled B. Letaief was supported in part by Hong Kong Research Grants Council under Grant 16212120.

ABSTRACT This paper proposes a Gaussian mixture model (GMM) based access point (AP) clustering technique in cell-free massive MIMO (CFMM) communication systems. The APs are first clustered on the basis of large-scale fading coefficients, and the users are assigned to each cluster depending on the channel gain. As the number of clusters increases, there is a degradation in the overall data rate of the system, causing a trade-off between the cluster number and average rate per user. To address this problem, we present an optimization problem that optimizes both the upper bound on the average downlink rate per user and the number of clusters. The optimal number of clusters is intuitively determined by solving the optimization problem, and then grouping the APs and users. As a result, the computation expense is much lower than the current techniques, since the existing methods require evaluations of the network performance in multiple iterations to find the optimal number of clusters. In addition, we analyze the performance of both balanced and unbalanced clustering. Numerical results will indicate that the unbalanced clustering yields a superior rate per user while maintaining a lower level of complexity compared to the balanced one. Furthermore, we investigate the statistical analysis of the spectral efficiency (SE) per user in the clustered CFMM. The findings reveal that the SE per user can be approximated by the logistic distribution.

INDEX TERMS AP centric clustering, cell-free massive MIMO (CFMM), Gaussian mixture model (GMM), logistic distribution, spectral efficiency.

I. INTRODUCTION

THE investigation of using multiple antennas at the transmitter and receiver ends was prompted by the need for a larger data rate in wireless communications. Because of the gains in terms of spatial multiplexing and diversity, multiple-input multiple-output (MIMO) systems can provide more reliability and higher spectral efficiency (SE) than single-input single-output (SISO) systems. Massive MIMO systems, also referred to as large-scale antenna systems (LSASs), have been proposed in [1] to obtain additional gains and capacity. In massive MIMO systems, the base stations (BSs) are outfitted with a huge number of antennas. Massive MIMO systems have several benefits, which are illustrated in detail in [2]. But they also face many challenges, including difficulties in estimating the channel, signal processing, pre-coding methods, and pilot contaminations. Another major issue is inter-cell interference which will be even more critical for future wireless networks because of the network densification [3]. Each user in a traditional cellular network connects to the BS of one of the numerous cells (unless during handover). The channel capacity of cellular networks is suboptimal since improved SE (bps/Hz/user) may be attained by co-processing each signal at several access points (APs) [4]. The idea of signal co-processing is realized in [5] in a network-centric manner by separating APs into distinct clusters, which is further implemented as coordinated multipoint with joint transmission (CoMP-JT) [6]. Cell-free massive MIMO (CFMM) is a new idea proposed in [7] and [8] that comes from combining user centric connectivity, high-density distributed network structure, and time-division duplex (TDD) protocol. CFMM is a promising key technology for 6G communication networks as it provides better coverage and a uniform connection for all users [9].

In a CFMM system, a large number of geographically dispersed APs are linked to a central processing unit (CPU) and serve users cooperatively [7], [8], [9], [10]. A fronthaul connection is used to connect each AP to a CPU that is utilized for data processing and coordination purposes. In a traditional cell-free network, a small number of slowly moving users is served by a much larger number of APs, as proposed in [7] and [8]. All users send orthogonal pilot sequences synchronously so that the APs can estimate the channel state information. In [8], the number of available orthogonal pilots is considered to be much greater than the number of users due to the slow movement of users. The power optimization and linear pre-coding techniques in CFMM are evaluated in [8] and [11]. However, due to the limited duration of the coherence block, in general, the number of orthogonal pilots is smaller than the number of users, and various users need to use non-orthogonal pilot sequences [7]. The estimation of the channel is compromised by pilot signals transmitted by other users, which results in the pilot contamination effect. To handle non-orthogonal pilot symbols, random and greedy pilot assignments are proposed in [7]. A closed-form expression for the achievable rate of a cell-free network is derived in [7] and [10] which is further used for designing the max-min power control scheme. The signal processing of CFMM in a fully centralized, partially centralized, or fully localized way is discussed in [12].

It is practically unaffordable to coordinate a large number of widely spread APs to serve users due to the high complexity of the system and the requirement for wide-bandwidth fronthauls that can handle the data of all users. In order to decrease this fronthaul demand and computational cost, [13], [14], and [15] suggested a user centric method that restricts each AP's connection to a subset of users. Various unique clustered cell-free network topologies were subsequently presented in [16], [17], [18], [19], [20], [21], and [22], where the whole network is divided into a number of non-overlapping groups so that the users that are interfering with each other the most are contained within the same cluster and the inter-cluster interference is minimal.

The clustering of CFMM is basically performed in two ways: user centric and AP centric. User centric clustered CFMM systems are formed by applying k-Means [16], k-Means++, improved k-Means++ [17], agglomerative hierarchical clustering [18] to cluster the users, followed by different AP selection methods. On the other hand, AP centric clustered CFMM systems use deep reinforcement learning [19] or weighted bipartite graphs [20], [21] for AP groupings. But the main concern is how to decide the optimal number of clusters to be formed and what should be the deciding parameters. Optimization problems can be formulated to find the optimal number of clusters that maximizes the sum throughput or maximizes the 95%-likely throughput [23]. Two techniques are discussed in [22]: one is fronthaul optimization with a minimum per user signal-to-interference-plus-noise ratio (SINR) constraint, and the second is max-min SINR optimization with a cluster size constraint. However, as mentioned in [22], the computation time for finding a solution for these optimization problems increases as the size of the network grows. To maximize the number of clusters with a per user rate constraint, a rate-constrained network decomposition (RC-NetDecomp) algorithm was proposed in [20]. However, the overall computational complexity of the RC-NetDecomp algorithm is $\mathcal{O}(M^3 \log_2 M)$ [21] where M is the number of APs.

In this paper, we propose Gaussian mixture model (GMM) based AP clustering algorithms. GMM clustering is more flexible than other unsupervised clustering techniques because it is a probabilistic algorithm. GMM offers a way to quantify uncertainty by calculating probabilities for each sample associated with each cluster. The utilization of posterior probabilities facilitates the integration of a new user into the CFMM network without the need for re-clustering. Clustering algorithms like k-Means and graph partitioning make hard assignments with no probabilistic output. Moreover, compared to k-Means clustering and graph partitioning, GMM is more flexible in cluster shape and less susceptible to outliers. All clustering methods generate unbalanced groups, but it is simple to balance the groups using GMM because its probabilistic estimates provide greater flexibility in cluster types. Other methods in the existing literature [20], [21], [22] are complicated enough, as they use multiple iterations to find the optimal number of clusters and evaluate the performance in each iteration. Here, we do not need to cluster multiple times to obtain the optimal number of clusters. Instead, we intuitively find it by solving the optimization problem and then group the APs and users. The main contributions in our work are as follows:

- To the best of our knowledge, GMM based AP centric clustered CFMM has not been studied yet. Here, the APs are clustered on the basis of large-scale fading coefficients, followed by user assignment to each cluster.
- To solve the issue of finding the optimal number of clusters, we formulate an optimization problem that maximizes the upper bound on the average per user rate in the downlink, as well as the number of clusters.
- We discuss the methods of forming both balanced and unbalanced clusters. The unbalanced clustering always outperforms the balanced one in terms of SE per user. Moreover, creating balanced clusters and adding an equal number of APs and users to each cluster is sometimes unachievable. Hence, to find the optimal number of clusters, we assume that all the clusters have the same number of users and APs and solve the optimization problem.

- We evaluate the performance of the proposed clustered CFMM system on the basis of average rate, SE, and computational complexity. It is shown that the average rate is improved while reducing the complexity by least half of that of existing clustered CFMM structures in [16] and 1.6 times of that of [20] and [21].
- We also approximate the distribution of SE per user and observe that the SE per user almost follows a logistic distribution. The statistical parameters of the logistic distribution of the SE are further analyzed and numerically estimated.

The paper is organized as follows. Section II describes the system architecture. Section III gives an elaborated idea of the proposed methodology. The numerical results obtained and the variation of performance with different parameters are described in Section IV. In Section V, the statistics of the SE per user are analyzed and approximated by the logistic distribution. Section VI presents the comparison of our proposed clustered CFMM with existing clustering techniques. Finally, Section VII concludes the paper.

II. SYSTEM MODEL

Consider a cell-free MIMO system with K users and M APs where $M \gg K$. The APs are connected to a CPU using fronthaul links. The users and APs follow the TDD protocol, considering channel estimation from pilot signals followed by data transmission. The time duration for mutually orthogonal pilot symbols is taken as τ_p and the coherence time as τ_c. We consider the downlink transmission for this work. The channel between user k and AP m is taken as g_{km}, and the channel coefficient is represented as

$$g_{km} = \sqrt{\beta_{km}} h_{km}, \tag{1}$$

where β_{km} is the large-scale fading gain caused by shadowing and path loss and h_{km} represents the small-scale fading gain between user k and AP m. It is assumed that the gains h_{km} $\forall k, m$ are independent and identically distributed (i.i.d.) complex Gaussian random variables each with zero mean and unit variance. The received signal at user k for conventional CFMM is given by

$$y_k^{dl} = \sum_{m=1}^{M} g_{km}^{*} \sum_{i=1}^{K} w_{im} s_i + n_k = \mathbf{g}_k^H \sum_{i=1}^{K} \mathbf{w}_i s_i + n_k \ , \tag{2}$$

where s_k is the unit-power independent data signal for user k with $\mathbb{E}\{|s_k|^2\} = 1$, n_k is the additive white Gaussian noise with mean zero and variance N_0, $\mathbf{g}_k = [g_{k1}, \cdots, g_{kM}]^T$ is the complex channel vector for the kth user, and $\mathbf{w}_k = [w_{k1}, \cdots, w_{kM}]^T$ is the precoding vector for user k. Here $(\cdot)^*$ denotes the complex conjugate, $(\cdot)^T$ denotes the transpose, $(\cdot)^H$ denotes the complex conjugate transpose or Hermitian, and $\mathbb{E}\{\cdot\}$ denotes the expectation operator. For the conventional maximum ratio (MR) precoding, w_{km} is calculated as

$$w_{km}^{MR} = \sqrt{\rho_k} \frac{\widehat{g}_{km}}{\sqrt{\mathbb{E}\{|\widehat{g}_{km}|^2\}}}, \tag{3}$$

where $\rho_k \geq 0$ is the transmit power allocated to user k and $\widehat{g}_{km}$ is the estimated channel between user k and AP m. Pilot sequences are used to estimate the channel coefficient as $\widehat{g}_{km}$.

We consider that each user is served by only a group of APs. Let $d_{kl} = 1$ if AP l serves user k and $d_{kl} = 0$ otherwise. Hence, the received signal at user k as mentioned in (2) is modified for clustered CFMM as

$$y_k^{dl} = \sum_{m=1}^{M} g_{km}^{*} \sum_{i=1}^{K} d_{im} w_{im} s_i + n_k = \mathbf{g}_k^H \sum_{i=1}^{K} \mathbf{D}_i \mathbf{w}_i s_i + n_k \ , \tag{4}$$

where $\mathbf{D}_k = \text{diag}(d_{k1}, \cdots, d_{kM})$ is a diagonal matrix of size $M \times M$.

From [12], we know that the fully centralized minimum mean-square error (MMSE) processing gives better performance than MR or local processing. Hence, we shall use centralized MMSE processing. The MMSE combining vector is given by

$$\mathbf{v}_k = p_k \left(\sum_{i=1}^{K} p_i \mathbf{D}_k \widehat{\mathbf{g}}_i \widehat{\mathbf{g}}_i^H \mathbf{D}_k + \mathbf{Z}_k \right)^{\dagger} \mathbf{D}_k \widehat{\mathbf{g}}_k \ , \tag{5}$$

where $\mathbf{Z}_k$ is given in [15, eq. (18)], $(\cdot)^\dagger$ denotes the pseudo inverse of a matrix, p_k is the transmit power of user k for the uplink transmission, and $\widehat{\mathbf{g}}_k = [\widehat{g}_{k1}, \cdots, \widehat{g}_{kM}]^T$ is the estimated channel vector of the kth user.

Motivated by the uplink-downlink duality, we select the downlink precoding vectors as

$$\mathbf{w}_i = \sqrt{\rho_i} \bar{\mathbf{w}}_i, \tag{6}$$

where

$$\bar{\mathbf{w}}_i = \frac{\mathbf{v}_i}{\sqrt{\mathbb{E}\{\mathbf{v}_i^H \mathbf{D}_i \mathbf{v}_i\}}}. \tag{7}$$

The downlink SE per user is calculated as

$$SE_k^{(dl)} = (1 - \frac{\tau_p}{\tau_c}) R_k \tag{8}$$

$$= (1 - \frac{\tau_p}{\tau_c}) \log_2 \left(1 + \Gamma_k^{(dl)} \right) , \tag{9}$$

where R_k is the achievable rate of user k and $\Gamma_k^{(dl)}$ is the SINR for user k, which is given by

$$\Gamma_k^{(dl)} = \frac{\rho_k \left| \mathbb{E} \left\{ \mathbf{g}_k^H \mathbf{D}_k \bar{\mathbf{w}}_k \right\} \right|^2}{\sum\limits_{\substack{i=1 \\ i \neq k}}^{K} \rho_i \mathbb{E}\left\{ \left| \mathbf{g}_k^H \mathbf{D}_i \bar{\mathbf{w}}_i \right|^2 \right\} + N_0} . \tag{10}$$

The average rate per user in the system is

$$R = \frac{1}{K} \sum_{k=1}^{K} R_k . \tag{11}$$

From [18, eq. (10)] we can calculate the ergodic average rate per user as

$$\bar{R} \triangleq \mathbb{E}_{\{\beta_{km}\}} \left[\mathbb{E}_{\mathbf{H}^H} \left[R_k \right] \right], \tag{12}$$

where $\mathbf{H} \in \mathbb{C}^{M \times K}$ is the small-scale fading matrix.

III. AP CENTRIC CLUSTERING

In a conventional CFMM structure, each AP serves all the users within the coverage area. Let $\mathbf{C}$ be the connectivity matrix representing the connections between APs and users. The size of $\mathbf{C}$ is taken as $M \times K$ where the element of the mth row and kth column is denoted by c_{mk}. For conventional CFMM we have

$$c_{mk}^{CFMM} = 1 \; \forall m \in \{1, \ldots, M\}, k \in \{1, \ldots, K\}. \quad (13)$$

The original CFMM system lacks scalability. Thus, user centric CFMM systems are proposed to reduce the fronthaul power consumption and computation load on the CPU. In [14], user k is associated with only $M_{0,k} \leq M$ APs using the largest large-scale fading based selection (LLSF) method. For the user centric approach, the group of APs serving user k is denoted by $\mathcal{M}_k^{uc}$ which is calculated as

$$\mathcal{M}_k^{uc} = \left\{ m \left| \frac{\sum_{m=1}^{M_{0,k}} \tilde{\beta}_{km}}{\sum_{m'=1}^{M} \beta_{km'}} \geqslant \varepsilon \right. \right\}, \quad (14)$$

where $\{\tilde{\beta}_{km}, \cdots, \tilde{\beta}_{kM}\}$ is the sorted set of the set $\{\beta_{km}, \cdots, \beta_{kM}\}$ in descending order and $0 \leq \varepsilon \leq 1$ is the predefined threshold. The connectivity matrix can then be formed as

$$c_{mk}^{UC} = \begin{cases} 1 & \text{if } m \in \mathcal{M}_k^{uc}, \\ 0 & \text{else.} \end{cases} \quad (15)$$

To address the scalability issue, one of the most commonly used user centric methods is dynamic cooperation clustering (DCC) [15], where a master AP is assigned to each user. This would help to form the dynamic cluster of that particular user by cooperating with APs nearer to the user.

The AP subsets obtained by user centric CFMM overlap with each other, which results in intra-cluster pilot contamination. As such, different user centric [16], [17], [18] and AP centric clustering [19], [20], [21] methods were proposed to form non-overlapping groups so that interference cancellation can be applied in each groups independently. In user centric clustering, the users are clustered first, and then different AP selection algorithms are applied to decide which AP belongs to which cluster. In AP centric clustering, the APs are grouped first, and then the users are assigned to the cluster where their associated AP belongs.

Next we discuss AP centric clustering using the GMM technique to form disjoint groups on the basis of large-scale fading.

A. GMM CLUSTERING

Clustering is an unsupervised machine learning technique. From a statistical perspective, clustering techniques may be separated into non-parametric and probabilistic model-based approaches. Some well-known non-parametric clustering techniques are k-Means, fuzzy c-Means (FCM), and hierarchical clustering. GMM is a probability model-based approach that uses the expectation maximization (EM) algorithm to calculate the mixture likelihood [24], [25]. Let the mean vector, covariance matrix, and weight of the lth Gaussian distribution be $\boldsymbol{\mu}_l, \boldsymbol{\Phi}_l, \pi_l$, respectively. The Gaussian parameters for all L Gaussian distributions are represented by $\Theta = \left\{\boldsymbol{\mu}_l, \boldsymbol{\Phi}_l, \pi_l | \forall l \in \{1, \ldots, L\}\right\}$. Firstly, the random Gaussian parameters (Θ) are taken as the initial starting point. An iterative algorithm is then used, as described below, until convergence.

- Expectation step:
  For a given Θ, compute the responsibilities of the mth sample (the posterior probability of the lth Gaussian distribution given a data point m) as
  $$p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l) = \frac{\pi_l p(\mathbf{x}_m | z_m = l, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)}{\sum_{l=1}^{L} \pi_l p(\mathbf{x}_m | z_m = l, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)}, \quad (16)$$
  where $z_m = l$ denotes that the mth sample belongs to the lth Gaussian distribution and $p(\mathbf{x}_m | z_m = l, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)$ is the likelihood of observing the mth sample given that it came from Gaussian l; it is given by
  $$p(\mathbf{x}_m | z_m = l, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l) = \mathcal{N}(\mathbf{x}_m | \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l), \quad (17)$$
  where $\mathcal{N}(\cdot)$ represents the Gaussian distribution.
- Maximization step:
  Θ is updated by maximizing the expected complete log likelihood given by
  $$\begin{aligned} &\max_{\Theta} \mathbb{E}[\ln(p(\mathbf{x}, z | \Theta))] \\ &= \max_{\Theta} \sum_{m=1}^{M} \sum_{l=1}^{L} p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l) \\ &\qquad \left(\ln \pi_l + \ln \mathcal{N}(\mathbf{x}_m | \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)\right). \quad (18) \end{aligned}$$
- Parameter update:
  The estimated parameters in each iteration are given by
  $$\hat{\pi}_l = \frac{\sum_{m=1}^{M} p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)}{M}, \quad (19a)$$
  $$\hat{\boldsymbol{\mu}}_l = \frac{\sum_{m=1}^{M} p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l) \mathbf{x}_m}{\sum_{m=1}^{M} p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)}, \quad (19b)$$
  $$\hat{\boldsymbol{\Phi}}_l = \frac{\sum_{m=1}^{M} p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)(\mathbf{x}_m - \boldsymbol{\mu}_l)(\mathbf{x}_m - \boldsymbol{\mu}_l)^T}{\sum_{m=1}^{M} p(z_m = l | \mathbf{x}_m, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l)}. \quad (19c)$$
  If the parameters do not converge, then the parameters in each step are updated as follows
  $$\{\pi_l, \boldsymbol{\mu}_l, \boldsymbol{\Phi}_l\} \leftarrow \{\hat{\pi}_l, \hat{\boldsymbol{\mu}}_l, \hat{\boldsymbol{\Phi}}_l\}.$$

The k-Means clustering algorithm can be seen as a specific instance of GMM where the covariance matrix is set as a scaled identity matrix. The GMM algorithm has demonstrated its superiority over the k-Means algorithm by effectively identifying clusters with arbitrary ellipsoidal shapes, regardless of the number of data points within each cluster [26]. Additionally, the weights are defined in a way such that just one element has a weight of 1, while all other elements have weights of 0. This leads to a hard assignment for k-Means.

B. GMM BASED CLUSTERED CFMM

The clustering technique proposed here basically ensures that the orthogonal pilots are preferably reused by users from different clusters. GMM clustering has not been used in the CFMM structure so far. We will be using GMM to divide the APs having a similar large-scale fading into L disjoint clusters. The main bottleneck of the clustered CFMM system is L, which must be fixed beforehand.

Once the number of clusters is fixed, we can easily divide the APs and the users into disjoint groups depending upon their large-scale channel gains. It is assumed that the large-scale channel gain is perfectly known to the CPU. Let the large-scale fading vector for AP m be represented by $\mathbf{b}_m = [\beta_{1m}, \cdots, \beta_{Km}]^T$. We will then form L disjoint clusters of M APs where the lth cluster is termed as $\mathcal{M}_l$. The number of APs in $\mathcal{M}_l$ is denoted by M_l and $\sum_{l=1}^{L} M_l = M$. Each cluster has a centroid $\boldsymbol{\mu}_l$ $(\forall l = 1, \ldots L)$. After clustering the APs, each user joins the cluster whose centroid is closest. The centroid $\boldsymbol{\mu}_l$ is a $K \times 1$ vector indicating propagation losses from each user to the centroid of cluster l. The group of users served by cluster l is termed as κ_l and is computed as

$$\kappa_l = \{\text{user } k \text{ served by cluster } l \text{ if } \boldsymbol{\mu}_l(k) > \boldsymbol{\mu}_z(k)\} \\ \forall z \neq l\ , z \in \{1, \ldots, L\} \text{ and } k \in \{1, \ldots, K\}, \tag{20}$$

where $\boldsymbol{\mu}_l(k)$ represents the kth element of the vector $\boldsymbol{\mu}_l$. So the connectivity matrix can be formed as

$$c_{mk} = \begin{cases} 1 & \text{if user } k \in \kappa_l \text{ and AP } m \in \mathcal{M}_l, \\ 0 & \text{else.} \end{cases} \tag{21}$$

The GMM based clustering method can be summarized by Algorithm 1.

Now the main problem is to decide the optimal number of clusters, i.e., L. There is a trade-off between the number of clusters and the sum rate of the system [19]. In the next subsection we formulate an optimization problem with some assumptions to find the optimal value of L which maximizes the overall rate as well as the scalability of the system.

Algorithm 1 GMM Based Clustered CFMM

```
Input: $\beta_{km} \forall k, m, L$
Output: $\mathcal{M}_l, \kappa_l, \boldsymbol{\mu}_l, \mathbf{C}$
1. Compute the large-scale fading vectors $\mathbf{b}_1, \ldots, \mathbf{b}_M$ for all APs.
2. Apply GMM clustering to divide $M$ APs into $L$ clusters to obtain $\mathcal{M}_1, \ldots, \mathcal{M}_L$ with their centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L$, respectively.
3. User assignment:
• for $k = 1$ to $K$ do
•     user $k \in \kappa_l$ if $\boldsymbol{\mu}_l(k) > \boldsymbol{\mu}_z(k)$ $\forall z \neq l$ , $z \in \{1, \ldots, L\}$
• end for
4. Calculation of C:
• for $m = 1$ to $M$ do
•     $\tilde{l} \leftarrow$ cluster of AP $m$
•     for $k = 1$ to $K$ do
•         if user $k \in \kappa_{\tilde{l}}$ then $c_{mk} = 1$
•         else $c_{mk} = 0$
•         end if
•     end for
• end for
```

C. OPTIMAL NUMBER OF CLUSTERS

The theoretical analysis of the rate performance for clustered CFMM is still unknown due to the lack of effective modeling of inter cluster interference [18]. Here two assumptions are made, which intuitively help to determine the number of clusters. The first assumption is related to the number of users in each cluster which is assumed to be the same $(K_l = K/L, \forall l)$, i.e., we consider L balanced clusters. The second assumption is that the ratio of the number of APs to the number of users in each cluster is fixed and denoted as λ. From [18, eq. (18)] we can write the upper bound of the average per user rate which is a function of L for large M and balanced clusters as

$$\bar{R} \leq \bar{R}^{ub} = F_1(L), \tag{22}$$

$$F_1(L) = \frac{\alpha}{2}\left[\log_2\left((\lambda-1)\frac{K}{L}+1\right) - \log_2\left(\frac{\lambda K}{M}\right) + \gamma \log_2 e\right], \tag{23}$$

where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant, α is the path-loss exponent, and the AP-selection ratio λ satisfies $1 < \lambda \leq M/K$. We consider $\alpha = 3.76$.

The value of $\bar{R}^{ub}$ is maximum when there are no clusters, i.e., $L = 1$, and gradually decreases as the number of clusters increases. To maintain scalability, more clusters are desirable. If an AP serves several users per pilot symbol, then the signals to and from these pilot-sharing users would overlap significantly, which is undesirable. When L increases, the complexity of the channel estimation and signal processing (i.e., precoding and combining) becomes fixed and scalable. The fronthaul links need to handle τ_p parallel uplink and downlink data signals per AP and each AP preferably serves at most one user per pilot symbol if $K_l \leq \tau_p$.

We cannot separately optimize $F_1(L)$ and L because $F_1(L)$ is dependent on the number of clusters. It is necessary to consider a joint optimization function that incorporates both the average per user rate and the number of clusters. The relationship between $F_1(L)$ and L can be fairly characterized by utilizing the cost function $F(L)$, which is given by

$$F(L) = F_1(L)F_2(L) = F_1(L)\left(1 - \frac{1}{L}\right). \tag{24}$$

The cost function $F(L)$ maximizes both the upper bound and the number of clusters.[1] The clarification about the optimization function is elaborated on in Appendix A. The optimization problem becomes

$$\max_{L \in \mathbb{N}} F(L)\,, \tag{25}$$

where $\mathbb{N}$ denotes the set of natural numbers.

We obtain the optimal number of clusters by solving the equation

$$\frac{\partial F(L)}{\partial L} = F_1(L)F_2(L)' + F_1(L)'F_2(L) = 0\,, \tag{26}$$

which can also be solved as

$$F_1(L)\frac{1}{L^2} + \frac{\alpha \log_2 e}{2}\left(\frac{(\lambda-1)K}{(\lambda-1)\frac{K}{L}+1}\right)\left(\frac{-1}{L^2}\right)F_2(L)$$
$$= 0 \tag{27}$$
$$\implies F_1(L) = \frac{\alpha \log_2 e}{2}\left(\frac{(\lambda-1)K}{(\lambda-1)\frac{K}{L}+1}\right)F_2(L) \tag{28}$$
$$\implies \log_2\left((\lambda-1)\frac{K}{L}+1\right) - \log_2\left(\frac{\lambda K}{M}\right) + \gamma \log_2 e$$
$$= \log_2 e \frac{(\lambda-1)K}{(\lambda-1)\frac{K}{L}+1}\left(1-\frac{1}{L}\right). \tag{29}$$

The optimal number of clusters, L_{opt}, satisfies

$$(\lambda-1)\frac{K}{L_{opt}} + 1 = \frac{\lambda K}{M e^{\gamma}} e^{\frac{(\lambda-1)K(L_{opt}-1)}{(\lambda-1)K+L_{opt}}}\,. \tag{30}$$

If the solution is not an integer, then we round it to the nearest positive integer to obtain L_{opt}.

Algorithm 2 shows how to obtain balanced clusters using GMM clustering where the APs in the lth cluster are grouped as $\widetilde{\mathcal{M}_l}$ and the users in the lth cluster are grouped as $\widetilde{\kappa_l}$. By solving (30), we get L_{opt} which is used to form L_{opt} number of balanced clusters using Algorithm 2. Initially, the AP clustering is done using the GMM clustering technique, which creates unbalanced clusters. We then balance the clusters forcefully by assigning $M_l = M/L_{opt}$ number of APs in each cluster. Step 4 of Algorithm 2 shows how the excess $M_l - M/L_{opt}$ elements in the lth cluster are shifted to the clusters having the second highest posterior probability $P_{m,l}$. One advantage of using GMM clustering is that it returns the posterior probability ($P_{m,l}$), that is, the probability of cluster l given a data point m. The probability $P_{m,l}$ of each $l \in \{1, \ldots, L_{opt}\}$ given each observation $m \in \{1, \ldots, M\}$ is calculated using Bayes' theorem as mentioned in (16). The values of $\boldsymbol{\mu}_l, \boldsymbol{\Phi}_l, \pi_l\ \forall l \in \{1, \ldots, L_{opt}\}$ are obtained from the maximization step after convergence. Similarly, the excess users are also forced to be shifted to the cluster having the second lowest propagation loss as given in Step 8 of Algorithm 2. Once the balanced AP clustering and user assignment are completed, the $\mathbf{C}$ matrix is calculated just like in Step 4 of Algorithm 1.

Algorithm 2 GMM Based Balanced Clustered CFMM

```
Input: $\beta_{km} \forall k, m, L_{opt}$
Output: $\widetilde{\mathcal{M}_l}, \widetilde{\kappa_l}, \boldsymbol{\mu}_l, \mathbf{C}$
1. Compute the large-scale fading vectors $\mathbf{b}_1, \ldots \mathbf{b}_M$ for all APs.
2. Apply GMM clustering to divide $M$ APs into $L$ number of clusters to obtain $\mathcal{M}_1, \ldots, \mathcal{M}_{L_{opt}}$.
3. Take a cluster set $\mathcal{G}$ whose APs are not balanced yet. Initially $\mathcal{G} = \{1, \ldots, L_{opt}\}$.
4. Balancing based on posterior probability $P_{m,l}$:
for $l = 1$ to $L_{opt} - 1$ do
    if $M_l > M/L_{opt}$ then
        Shift $M_l - M/L_{opt}$ APs to the cluster $l'$ such that $P_{m,l} > P_{m,l'} > P_{m,q}\ \forall q \neq l, l',\ q \in \mathcal{G}$
        $\mathcal{G} \leftarrow \mathcal{G} - \{l\}$
    end if
end for
5. Now we get balanced AP clusters $\widetilde{\mathcal{M}_1}, \ldots, \widetilde{\mathcal{M}_{L_{opt}}}$ and compute the means to obtain their centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{L_{opt}}$.
6. User assignment:
for $k = 1$ to $K$ do
    user $k \in \kappa_l$ if $\boldsymbol{\mu}_l(k) > \boldsymbol{\mu}_z(k)\ \forall z \neq l\ , z \in \{1, \ldots, L_{opt}\}$
end for
7. Again take a cluster set $\mathcal{G}$ whose users are not balanced yet. Initially $\mathcal{G} = \{1, \ldots, L_{opt}\}$.
8. Balancing based on propagation loss:
for $l = 1$ to $L_{opt} - 1$ do
    if $K_l > K/L_{opt}$ then
        Shift $K_l - K/L_{opt}$ users to the cluster $l'$ such that $\boldsymbol{\mu}_l(k) > \boldsymbol{\mu}_{l'}(k) > \boldsymbol{\mu}_q(k)\ \forall q \neq l, l',\ q \in \mathcal{G}$
        $\mathcal{G} \leftarrow \mathcal{G} - \{l\}$
    end if
end for
9. Now we get balanced user clusters $\widetilde{\kappa_1}, \ldots, \widetilde{\kappa_{L_{opt}}}$.
10. Calculation of $\mathbf{C}$:
for $m = 1$ to $M$ do
    $\tilde{l} \leftarrow$ cluster of AP $m$
    for $k = 1$ to $K$ do
        if user $k \in \widetilde{\kappa_{\tilde{l}}}$ then $c_{mk} = 1$
        else $c_{mk} = 0$
        end if
    end for
end for
```

IV. NUMERICAL RESULTS

We evaluate the performance of the clustered CFMM by analyzing its SE per user, 90% likely SE point, average SE per user, and average minimum SE. The quantity 90% likely SE point denotes the fairness [27] of the system, i.e., 90% of the users have an SE greater than that. Here M APs and K users are distributed in a 2×2 km^2 area. Both the users and the APs are equipped with one antenna each and are

[1] A more complicated cost function may yield better results than the proposed one. Determining the optimal cost function for enhancing scalability in CFMM remains an open problem.

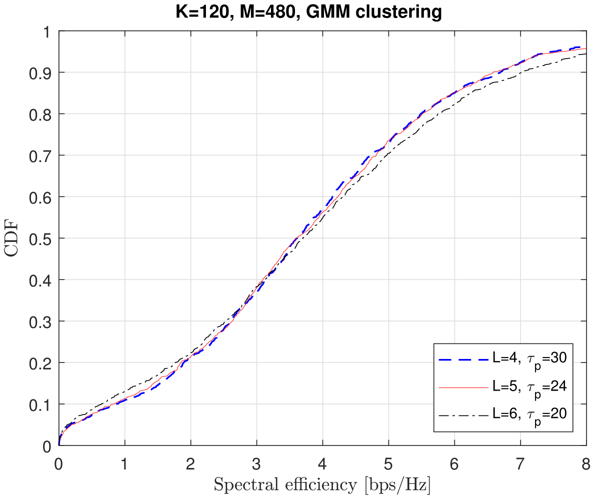

FIGURE 1. CDF of spectral efficiency per user with varying number of clusters for $K = 120$, $M = 480$; GMM clustering used to form balanced clusters.

distributed uniformly in a square coverage area. The other model parameters are $\tau_c = 200, p_k = 100\,\text{mW}, \rho_k = 1/\tau_p$ W, and a bandwidth of 20 MHz. The SE for each user is obtained numerically by considering 50 different set ups with 1000 channel realizations for each set up. The numerical findings are obtained using MATLAB, which, by default, uses the k-Means++ algorithm to initialize cluster centers for both GMM and k-Means clustering.

We consider $K = 120$ and $M = 480$ in Fig. 1 and use GMM clustering to obtain 4, 5, and 6 balanced clusters. For a fair comparison, we consider $\tau_p = 30$ for $L = 4$, $\tau_p = 24$ for $L = 5$, and $\tau_p = 24$ for $L = 6$. Fig. 1 shows the cumulative distribution function (CDF) of the downlink SE per user using MMSE processing. The 90% likely SE points for 4, 5, and 6, respectively, are $0.8730, 0.8444, \text{and}\, 0.6459$ bps/Hz. It is clear that as the number of clusters increases, the 90% likely SE point shifts left, indicating a decrease in user fairness. The 90% likely SE point 0.8730 implies that 90% of the users achieve 0.8730 bps/Hz SE per user. The average SE per user for $L = 4, 5, \text{and}\, 6$, respectively, are $3.7693, 3.7836, \text{and}\, 3.8808$ bps/Hz. The sum SE for $L = 4, 5, \text{and}\, 6$, respectively, are $452.3154, 454.0297$, and 465.7001 bps/Hz. The total SE of the system is increased by increasing the number of clusters and using $\tau_p = \lceil K/L \rceil$.

Fig. 2 shows the variation of the cost function $F(L)$ in (24) and the upper bound of the average per user rate $F_1(L)$ in (23) with respect to L. Here $K = 120$, $M = 480$, and $\lambda = 4$. The function $F_1(L)$ decreases as L increases. The cost function is concave in nature and has its maximum at $L = 6$.

Fig. 3 shows the CDF of the downlink SE per user for $K = 120$, $M = 480$, $\tau_p = 20$, and $L = 6$. Here we consider both unbalanced and balanced clusters formed by using Algorithms 1 and 2, respectively. It is clear from Fig. 3 that the SE performance of an unbalanced cluster is better than that of a balanced one, as the 90% likely SE points are 1.2685 and 1.2647, respectively, for GMM and k-Means clustering. It can be shown from Table 1 that both the average and the median SE per user of unbalanced GMM clustering is better than those of the balanced clustering, as well as the unbalanced k-Means clustering.[2] The unbalanced clusters are formed by grouping similar types of APs together and the user assignment is done on the basis of the least propagation loss from the centroid. But the balanced cluster was formed by forcefully shifting some APs from their original cluster to a cluster having the second highest posterior probability based on the vacancies in other clusters. Some users are also served by a cluster whose centroid is the second nearest, as the nearest centroid cluster is full. So the overall performance degrades. This observation emphasizes the correlation between cardinality balancing and error performance, as discussed in [28]. Moreover, it is always not possible to make a balanced cluster, i.e., assign an equal number of APs and users to each cluster. Hence, we will be using the optimal number of clusters obtained from (30) to form unbalanced clusters using Algorithm 1.

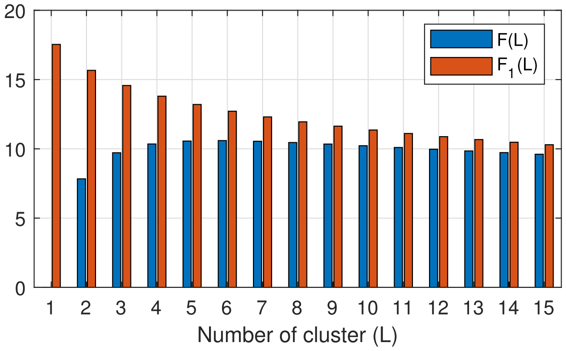

FIGURE 2. Variation of cost function and upper bound of average per user rate with L for $K = 120$, $M = 480$.

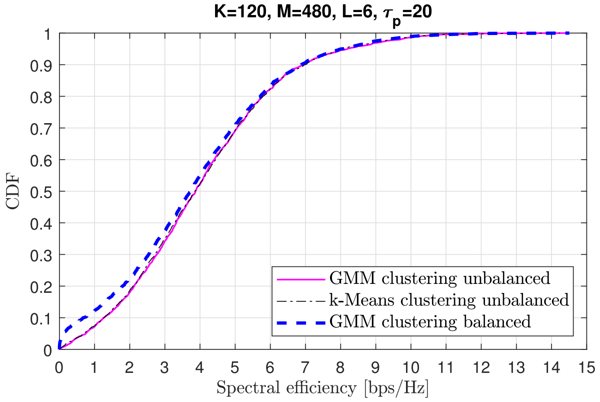

FIGURE 3. CDF of spectral efficiency per user for unbalanced and balanced clusters using GMM and k-Means clustering.

Figs. 4, 5, and 6 show the performance comparison of two different clustering methods, GMM and k-Means, that are used to form unbalanced clusters for $K = 100$, and $M = 400$. Fig. 4 shows how the average SE per user decreases as the

[2] The standard error for average SE for balanced GMM clustering is 0.0339, for unbalanced GMM clustering it is 0.0322, and for unbalanced k-Means clustering it is 0.0323, as shown in Table 1.

TABLE 1. Comparison of balanced and unbalanced clusterings for $K = 120$, $M = 480$, $L = 6$, $\tau_p = 20$.

Methods	Average SE	80% likely SE point	90% likely SE point	Median SE point
Balanced GMM clustering	3.8808	1.7925	0.6782	3.6994
Unbalanced GMM clustering	4.0563	2.1267	1.2685	3.8737
Unbalanced k-Means clustering	4.0389	2.0958	1.2647	3.8522

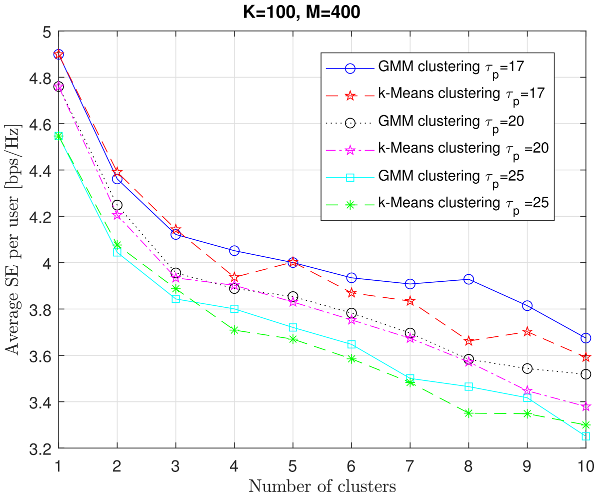

FIGURE 4. Average SE per user versus number of clusters for clustered CFMM using GMM and k-Means clustering.

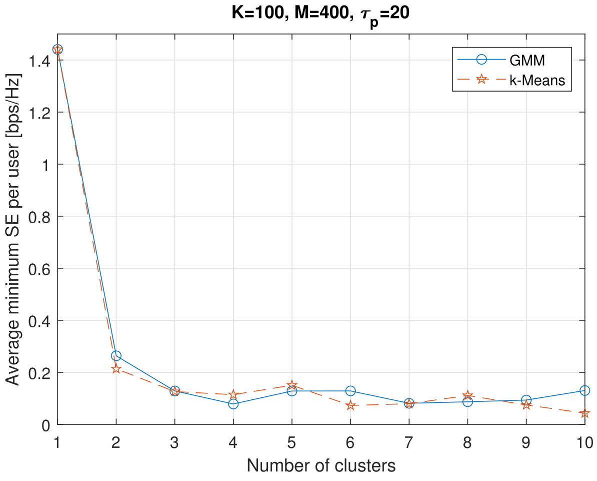

FIGURE 5. Average minimum SE versus number of clusters for clustered CFMM using GMM and k-Means clustering.

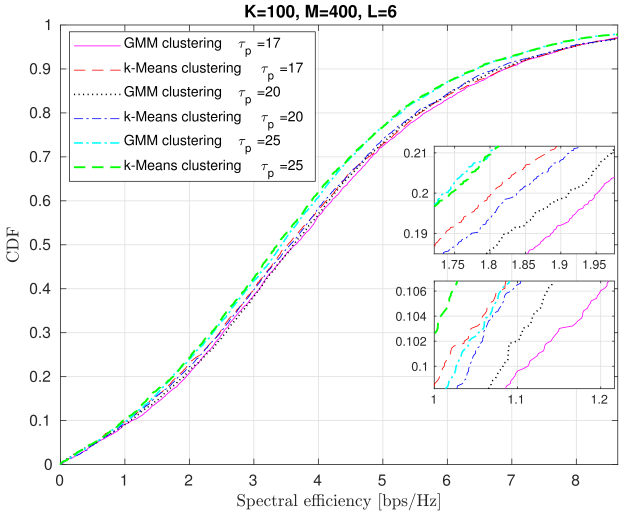

FIGURE 6. CDF of spectral efficiency per user with varying τ_p for $K = 100$, $M = 400$, and $L = 6$; GMM and k-Means clustering used to form unbalanced clusters.

number of clusters increases. It is also clear that the average SE per user decreases as τ_p increases. In general, GMM clusters offer a better average SE per user than k-Means clustering, but for large τ_p, k-Means can outperform GMM clustering. Fig. 5 shows how the average minimum SE starts to degrade abruptly when we start forming clusters. When there are no clusters, the CFMM system performs better, but this comes at a cost, such as higher fronthaul requirements for data co-processing and longer delays.

Fig. 6 shows the variation in the spectral efficiency per user with respect to τ_p and the number of clusters. If we fix $L = 6$, the minimum number of orthogonal pilots required is $\lceil K/L \rceil = 17$. Due to unbalanced clustering, some clusters may require $\tau_p > 17$. It is clear from Fig. 6 and Table 2 that as τ_p increases from 17 to 25, the 90% likely SE point decreases from 1.1128 to 1.0263 for GMM clustering. The 80% likely SE point, 90% likely SE point, and the median shift towards the left indicating a decrease in the overall rate as we increase τ_p, as shown in Table 2. Further incrementing τ_p does not help much to improve the system performance as the $\max(K_l) \leq 20\, \forall l \in \{1, \ldots, 6\}$ in case of GMM clustering and the $\max(K_l) \leq 23\, \forall l \in \{1, \ldots, 6\}$ in case of k-Means clustering. Hence, the best choice of τ_p is 17, namely, $\lceil K/L \rceil$. The performance of clustered CFMM for varying τ_p is tabulated in Table 2.[3] The total SE for GMM clustering is 7.45 bps/Hz (1.96%) higher than that for k-Means for $\tau_p = 17$. The total SE for GMM clustering is 4.91 bps/Hz (1.30%) higher than that for k-Means for $\tau_p = 20$. The total SE for GMM clustering is 2.95 bps/Hz (0.82%) higher than that for k-Means for $\tau_p = 25$. Table 2 shows that the 90% likely SE point for GMM clustering is improved by 9.94%, 3.84%, and 5.86% for $\tau_p = 17, 20,$ and 25, respectively, when compared

[3] The standard errors for average SE for GMM clustering, as shown in Table 2, are 0.0314, 0.0312, and 0.0300 for $\tau_p = 17, 20,$ and 25, respectively. The standard errors for average SE for k-Means clustering, as shown in Table 2, are 0.0318, 0.0312, and 0.0301 for $\tau_p = 17, 20,$ and 25, respectively.

TABLE 2. Comparison of unbalanced clustering for $K = 100$, $M = 400$, and $L = 6$.

τ_p	Methods	Average SE	80% likely SE point	90% likely SE point	Median SE point
17	GMM clustering	3.8710	1.95221	1.1128	3.6404
	k-Means clustering	3.7965	1.80283	1.0121	3.5257
20	GMM clustering	3.8330	1.91152	1.0825	3.6142
	k-Means clustering	3.7839	1.83933	1.0425	3.5632
25	GMM clustering	3.6231	1.73501	1.0263	3.4295
	k-Means clustering	3.5936	1.74482	0.9695	3.3636

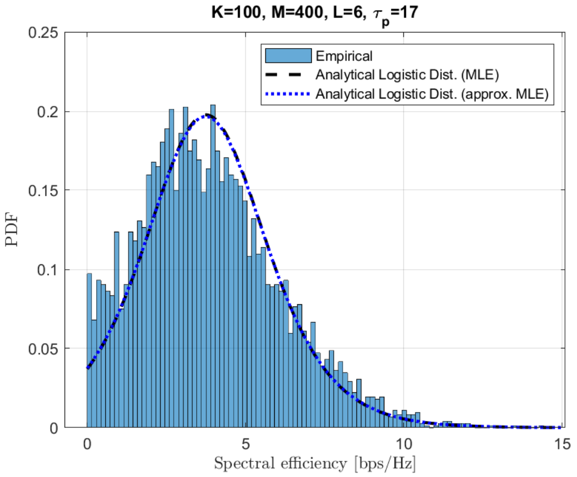

FIGURE 7. Empirical PDF and logistic approximation of spectral efficiency per user with GMM clustering for $K = 100$, $M = 400$, $L = 6$, and $\tau_p = 17$.

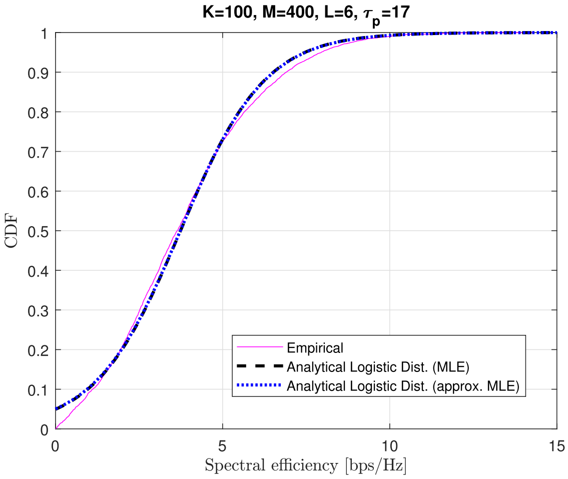

FIGURE 8. Empirical CDF and logistic approximation of spectral efficiency per user with GMM clustering for $K = 100$, $M = 400$, $L = 6$, and $\tau_p = 17$.

to k-Means. It is observed that as τ_p increases, the performance gap between GMM and k-Means clustering decreases.

V. APPROXIMATE ANALYSIS OF SE PER USER

Previously, Wang et al. have shown that the probability density function (PDF) of the mutual information in a distributed MIMO system can be approximated as a Gaussian distribution (for high SNR values) or a log-normal distribution (for low SNR values) in [29]. Hence, we numerically compute the SE per user for $K = 100$, $M = 400$, $L = 6$, $\tau_p = 17$ and estimate its statistics. It is observed that the SE per user follows a logistic distribution approximately, and its PDF is given by

$$g(x, \theta, s) = \frac{e^{\frac{x-\theta}{s}}}{s(1 + e^{\frac{x-\theta}{s}})^2}, \tag{31}$$

where θ is the location parameter and s is the scale parameter. The CDF of the logistic distribution is given by

$$G(x, \theta, s) = \frac{1}{1 + e^{-\frac{x-\theta}{s}}}. \tag{32}$$

Figs. 7 and 8 show the numerical logistic approximations to the SE PDF and CDF, respectively, with location parameter and scale parameter computed by the maximum likelihood (ML) estimation. The numerical estimation of the parameters gives a close match to the empirically generated PDF and CDF. The log-likelihood function for n samples is given by

$$\mathcal{L}(\theta, s) = n \ln(\frac{1}{s}) - \sum_{i=1}^{n} \frac{x_i - \theta}{s} - 2 \sum_{i=1}^{n} \ln(1 + e^{-\frac{x_i-\theta}{s}}). \tag{33}$$

The likelihood equations are obtained by

$$\frac{\partial \mathcal{L}(\theta, s)}{\partial \theta} = 0, \tag{34a}$$

$$\frac{\partial \mathcal{L}(\theta, s)}{\partial s} = 0. \tag{34b}$$

The ML estimates (MLEs) of the two parameters, $\hat{\theta}$ and $\hat{s}$, can be obtained by solving (34a) and (34b), respectively, which further implies

$$\hat{\theta} \sum_{i=1}^{n} \left[\frac{1}{e^{\frac{x_i-\hat{\theta}}{\hat{s}}} + 1} \right] = \frac{n}{2}, \tag{35a}$$

$$n\hat{s} = \sum_{i=1}^{n} \left[\left(x_i - \hat{\theta} \right) \tanh \left(\frac{x_i - \hat{\theta}}{2\hat{s}} \right) \right]. \tag{35b}$$

The likelihood equations (35a) and (35b) do not provide any direct solution, and these are to be solved iteratively using

TABLE 3. Empirical and approximate statistics of se per user for $K = 100$, $M = 400$, $\tau_p = 17$.

Methods	Location parameter	Scale parameter	Mean	Standard deviation	90% likely SE point	Median SE point
Empirical			3.8710	2.2252	1.1128	3.6404
Logistic distribution with MLE	3.7389	1.2642	3.7389	2.2929	0.9613	3.7389
Logistic distribution with approx. MLE	3.7346	1.2710	3.7346	2.3053	0.9419	3.7346

TABLE 4. Empirical and approximate statistics of se per user for $K = 100$, $M = 400$, $\tau_p = 20$.

Methods	Location parameter	Scale parameter	Mean	Standard deviation	90% likely SE point	Median SE point
Empirical			3.8330	2.2116	1.0825	3.6142
Logistic distribution with MLE	3.7045	1.2490	3.7045	2.2655	0.9601	3.7045
Logistic distribution with approx. MLE	3.7009	1.2554	3.7009	2.2771	0.9425	3.7009

TABLE 5. Empirical and approximate statistics of se per user for $K = 100$, $M = 400$, $\tau_p = 25$.

Methods	Location parameter	Scale parameter	Mean	Standard deviation	90% likely SE point	Median SE point
Empirical			3.6231	2.1225	1.0263	3.4295
Logistic distribution with MLE	3.4952	1.1945	3.4952	2.1666	0.8706	3.4952
Logistic distribution with approx. MLE	3.4904	1.2010	3.4904	2.1784	0.8516	3.4904

computers to obtain $\hat{\theta}$ and $\hat{s}$ [30]. In the iterative estimation process, each parameter is estimated one at a time in a cyclic order, skipping one parameter if it is assumed to be known. An initial estimate is to be selected first for the unknown parameters.

We can avoid the complex iterative techniques by using [31], which gives the approximate MLEs of the two parameters as

$$\hat{\hat{\theta}} = B - \hat{\hat{s}}C\,, \tag{36}$$

$$\hat{\hat{s}} = \frac{-D + \left(D^2 + 4nE\right)^{1/2}}{2n}\,, \tag{37}$$

where $B, C, D,$ and E are computed from the SE per user obtained from the clustered CFMM using (8). The calculation of the coefficients $B, C, D,$ and E are discussed in Appendix B.

The mean and median of the logistic distribution is θ and the standard deviation is $(s\pi/\sqrt{3})$. The 90% likely SE point, $SE_{90\%}$, is calculated by solving

$$G(SE_{90\%}, \theta, s) = \frac{1}{1 + e^{-\frac{SE_{90\%}-\theta}{s}}} = 0.1$$
$$\implies SE_{90\%} = \theta - s\ln(9)\,. \tag{38}$$

Tables 3, 4, and 5 show the sample mean, median, and standard deviation, respectively, of the SE per user obtained numerically, as well as the parameters estimated for the logistic approximation for $\tau_p = 17, 20,$ and 25. It is observed that the parameters obtained by the approximate MLEs are almost the same as the parameters obtained by MLEs, i.e., $\hat{\hat{\theta}} \approx \hat{\theta}$ and $\hat{\hat{s}} \approx \hat{s}$. It is observed from Tables 3, 4, and 5 that the estimated location parameter of the logistic approximation, $\hat{\hat{\theta}}$, also gives the upper limit of the median SE per user and the lower limit of the average SE per user for all τ_p. Similarly, the 90% likely SE point calculated from (38) presents the lower limit for the actual 90% likely SE point. To formulate an optimization problem that maximizes the 90% likely SE point, the 80% likely SE point, or the median, we can use the logistic distribution approximation presented in (38). Additionally, the statistics may be utilized as constraints of optimization problems. Focusing on maximizing the number of clusters while maintaining a 90% probable SE point constraint is one approach. Fronthaul optimization while maintaining a 90% probable SE point constraint is an alternative.

VI. COMPARISON WITH OTHER CLUSTERED CFMM

Our proposed method finds the optimal number of clusters in a single step by solving the optimization problem and directly performs the clustering using GMM. The complexity of GMM clustering is $\mathcal{O}(LMK^3)$. Another advantage is that a new user can be instantly added by using its large-scale fading coefficient. For a new user, we can use the parameters obtained from the existing clustering. We compute the mean for the newly added feature, i.e., $\beta_{(k+1)m} \forall m = 1, \ldots, M$ for all the L clusters. The mean for the $(K+1)$th feature is calculated as $\boldsymbol{\mu}_l(K+1) = \left(\sum_{m=1}^{M} P_{m,l}\beta_{(k+1)m}\right)/\left(\sum_{m=1}^{M} P_{m,l}\right)$

for clusters $l = 1, \ldots, L$ and then the centroid of each cluster is compared for user assignment. The new user is served by the AP whose centroid has the highest value, i.e. $\boldsymbol{\mu}_l(K+1) > \boldsymbol{\mu}_z(K+1)\ \forall z \neq l\ , z \in \{1, \ldots, L\}$. We can fix some threshold SE for the $(K+1)$th user to detect whether it is grouped correctly. If the SE of the $(K+1)$th user is less than the threshold, then the clustering of all the M APs and $K+1$ users need to be done again. Further analysis of adding a new user and deciding on the value of the threshold is not included in this work as it is beyond the scope of this paper.

In [16], initially, the number of clusters is fixed to $L_0 = \lceil K/\tau_p \rceil$ and k-Means clustering is performed until $K_l \leq \tau_p$, $\forall l = 1, \ldots, L$. This ensures that each user in a cluster is assigned to one orthogonal pilot. The optimal number of clusters is approximately $L_0 + \log_2(L_0)$ and to find it, k-Means might be performed up to $(L_0 - 1)$ times. So the overall complexity to cluster K users is $\mathcal{O}(TKML(L_0 - 1))$, with T being the number of iterations required to converge. This method is computationally faster as it uses k-Means clustering. But k-Means offers less flexibility than GMM in terms of the cluster size, and it is more difficult to form balanced clusters by using k-Means. Moreover, numerical results show that the SE per user is improved by using GMM clustering. Even if we use GMM clustering and follow the method proposed in [16], the clustering operation needs to be performed more than once to ensure its criteria.

In [18], agglomerative hierarchical clustering is employed to cluster the users into L groups, and it ensures that the AP-selection ratio λ is fixed, i.e., the number of APs serving each cluster is $M_l = \lambda K_l$. It is shown in [18] that as the number of clusters increases, both the average rate per user and the average minimum rate of users decrease. However, no criteria are discussed to find the optimal number of clusters. Agglomerative hierarchical clustering is slower than other clustering methods as the time complexity is $\mathcal{O}(K^3)$ and memory complexity is $\mathcal{O}(K^2)$. Agglomerative hierarchical clustering is less preferable for massive MIMO systems and is not flexible enough to assign a group to a new user.

In [20] and [21], binary searching is required to find the optimal number of clusters so that the average per-user rate is maintained above a threshold (R_{th}). Each time, the network has to perform an eigenvalue decomposition of the Laplacian matrix followed by k-Means clustering to find whether the average rate is above the threshold or not. To create L clusters of an L dimensional dataset, the complexity of k-Means is $\mathcal{O}(ML^2T)$, where T is the number of iterations required for convergence. In each search iteration, the complexity is $\mathcal{O}(ML^2T + M^3)$. The complexity of binary search is $\mathcal{O}(\log_2 M)$ and the overall complexity of RC-NetDecomp algorithm is $\mathcal{O}(M^3 \log_2 M)$ (as $M \gg L, T$). As mentioned in [20] for $K = 100$, $M = 200$, and $R_{th} = 3.5$, the optimal number of clusters is 9 and to find that optimal value the clustering operation is performed 8 times to ensure $R \geq R_{th}$. Thus, one can conclude that overall our proposed method has less complexity when the ratio of APs to users $M/K > \sqrt{K/\log_2 M}$. For $M = 400$ and $K = 100$, our proposed method is 1.6 times less complex than the method in [20] and [21] as $M/K > 3.33$. In addition, it will be necessary to partition the new graph after reconfiguring it if a new user joins the network.

VII. CONCLUSION

In this paper, we introduced a new application of GMM clustering to form unbalanced, as well as balanced clusters in CFMM. We discussed a method of obtaining the optimal number of balanced clusters by formulating an optimization problem that maximizes the upper bound on the average per user rate, as well as the number of clusters. Due to the relatively poor performance and impracticability of forming balanced clusters, we use the same optimization problem to obtain the optimal number of clusters to form unbalanced clusters. Numerical results demonstrated the superiority of GMM clustering over k-Means clustering in terms of average SE per user, median SE, and 90% likely SE point. We also evaluated the statistics of the SE per user and proposed that the SE can be numerically approximated by the logistic distribution.

APPENDIX A
CLARITY REGARDING THE OPTIMIZATION FUNCTION

The selection of the objective function (24) is determined by comparing cost functions of various functional forms. The choice of the $F_2(L)$ function is challenging as the cost function needs to be concave in nature. Here, $F_2(L)$ focuses on maximizing L while maintaining the concavity of $F(L)$. Examples of functions $F_2(L)$ that exhibit monotonic increase as L increases include L, $-1/L$, and $1 - 1/L$. Two potential cost functions that could be considered are $F_1(L) + F_2(L)$ or $F_1(L)F_2(L)$, as elaborated upon in the following discussion:

1) $F_1(L) + L$ is a convex function and maximizes at $L = \infty$;
2) $F_1(L) \times L$ is a monotonically increasing function and maximizes at $L = \infty$;
3) $F_1(L) - 1/L$ is a monotonically decreasing function and maximizes at $L = 1$;
4) $F_1(L) \times (-1/L)$ is a monotonically increasing function and maximizes at $L = \infty$;
5) $F_1(L) + 1 - 1/L$ is a monotonically decreasing function and maximizes at $L = 1$;
6) $F_1(L) \times (1 - 1/L)$ is a concave function and maximizes at $L = 6$ for $K = 100$ and $M = 400$, at $L = 5$ for $K = 10$ and $M = 100$, at $L = 4$ for $K = 50$ and $M = 100$, and other values of K and M as far as $K < M$. In a cell-free massive MIMO network a large number of geographically distributed APs simultaneously serve a small number of users and $M \gg K$ is generally considered. Thus, the cost function in (24) is a reasonable choice.

APPENDIX B
APPROXIMATE MLEs OF PARAMETERS OF LOGISTIC DISTRIBUTION OF SE

The coefficients of the approximate MLEs of the logistic distribution can be calculated from [31] and [32]. First we sort the n samples of SE per user such that $x_1' \leq \cdots \leq x_n'$ and compute the variables ν_i, ζ_i, b_i, a_i for $i = 1, \ldots, n$ as

$$\nu_i = \frac{i}{n+1}\,, \tag{39}$$

$$\zeta_i = \ln\left(\frac{\nu_i}{1-\nu_i}\right), \tag{40}$$

$$b_i = 2(1-\nu_i)\nu_i\,, \tag{41}$$

$$a_i = 1 - 2\nu_i + b_i\zeta_i\,. \tag{42}$$

Now we calculate the coefficients B, C, D, and E as

$$B = \frac{\sum_{i=1}^{n} b_i x_i'}{\sum_{i=1}^{n} b_i}\,, \tag{43}$$

$$C = \frac{\sum_{i=1}^{n} a_i x_i'}{\sum_{i=1}^{n} b_i}\,, \tag{44}$$

$$D = \sum_{i=1}^{n} a_i x_i' - BC \sum_{i=1}^{n} b_i\,, \tag{45}$$

$$E = \sum_{i=1}^{n} b_i (x_i')^2 - B^2 \sum_{i=1}^{n} b_i\,. \tag{46}$$

REFERENCES

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[2] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[3] G. Interdonato, E. Björnson, H. Quoc Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–13, Aug. 2019.

[4] S. Shamai (Shitz) and B. M. Zaidel, "Enhancing the cellular downlink capacity via co-processing at the transmitting end," in *Proc. IEEE VTS 53rd Veh. Technol. Conf., Spring*, Rhodes, Greece, May 2001, pp. 1745–1749.

[5] S. Zhou, M. Zhao, X. Xu, J. Wang, and Y. Yao, "Distributed wireless communication system: A new architecture for future public wireless access," *IEEE Commun. Mag.*, vol. 41, no. 3, pp. 108–113, Mar. 2003.

[6] R. Irmer et al., "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, Feb. 2011.

[7] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO: Uniformly great service for everyone," in *Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Stockholm, Sweden, Jun. 2015, pp. 201–205.

[8] E. Nayebi, A. Ashikhmin, T. L. Marzetta, and H. Yang, "Cell-free massive MIMO systems," in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2015, pp. 695–699.

[9] H. He, X. Yu, J. Zhang, S. Song, and K. B. Letaief, "Cell-free massive MIMO for 6G wireless communication networks," *J. Commun. Inf. Netw.*, vol. 6, no. 4, pp. 321–335, Dec. 2021.

[10] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[11] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4445–4459, Jul. 2017.

[12] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 77–90, Jan. 2020.

[13] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 706–709, Dec. 2017.

[14] H. Q. Ngo, L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 25–39, Mar. 2018.

[15] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[16] F. Riera-Palou, G. Femenias, A. G. Armada, and A. Pérez-Neira, "Clustered cell-free massive MIMO," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[17] Q. N. Le, V.-D. Nguyen, O. A. Dobre, N.-P. Nguyen, R. Zhao, and S. Chatzinotas, "Learning-assisted user clustering in cell-free massive MIMO-NOMA networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12872–12887, Dec. 2021.

[18] O. Zhou, J. Wang, and F. Liu, "Average downlink rate analysis for clustered cell-free networks with access point selection," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 742–747.

[19] C. F. Mendoza, S. Schwarz, and M. Rupp, "Deep reinforcement learning for spatial user density-based AP clustering," in *Proc. IEEE 23rd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Oulu, Finland, Jul. 2022, pp. 1–5.

[20] J. Wang, L. Dai, L. Yang, and B. Bai, "Rate-constrained network decomposition for clustered cell-free networking," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 2549–2554.

[21] J. Wang, L. Dai, L. Yang, and B. Bai, "Clustered cell-free networking: A graph partitioning approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5349–5364, Aug. 2023.

[22] C. F. Mendoza, S. Schwarz, and M. Rupp, "Cluster formation in scalable cell-free massive MIMO networks," in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Thessaloniki, Greece, Oct. 2020, pp. 62–67.

[23] E. Nayebi and B. D. Rao, "Access point location design in cell-free massive MIMO systems," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2018, pp. 985–989.

[24] D. J. Hand, G. J. McLachlan, and K. E. Basford, "Mixture models: Inference and applications to clustering," *Appl. Statist.*, vol. 38, no. 2, p. 384, 1989.

[25] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognit.*, vol. 45, no. 11, pp. 3950–3961, Nov. 2012.

[26] V.-E. Neagoe and V. Chirila-Berbentea, "Improved Gaussian mixture model with expectation-maximization for clustering of remote sensing imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Beijing, China, Jul. 2016, pp. 3063–3065.

[27] M. Farooq, H. Q. Ngo, E.-K. Hong, and L.-N. Tran, "Utility maximization for large-scale cell-free massive MIMO downlink," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 7050–7062, Oct. 2021.

[28] S. A. M. Anaraki and A. Haeri, "Soft and hard hybrid balanced clustering with innovative qualitative balancing approach," *Inf. Sci.*, vol. 613, pp. 786–805, Oct. 2022.

[29] D. Wang, J. Wang, X. You, Y. Wang, M. Chen, and X. Hou, "Spectral efficiency of distributed MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 10, pp. 2112–2127, Oct. 2013.

[30] H. L. Harter and A. H. Moore, "Maximum-likelihood estimation, from censored samples, of the parameters of a logistic distribution," *J. Amer. Stat. Assoc.*, vol. 62, no. 318, p. 675, Jun. 1967.

[31] N. Balakrishnan, "Approximate maximum likelihood estimation for a generalized logistic distribution," *J. Stat. Planning Inference*, vol. 26, no. 2, pp. 221–236, Oct. 1990.

[32] A. Asgharzadeh, R. Valiollahi, and M. Abdi, "Point and interval estimation for the logistic distribution based on record data," *SORT, Statist. Oper. Res. Trans.*, vol. 40, no. 1, pp. 89–112, Jan. 2016.

PIALY BISWAS (Member, IEEE) received the B.Tech. degree in electronics and telecommunication engineering from the National Institute of Technology Raipur, Raipur, in 2019, and the Ph.D. degree from the Bharti School of Telecommunication Technology and Management, Indian Institute of Technology (IIT) Delhi, New Delhi, in December 2023. She was an Early-Doc Fellow with IIT Delhi from December 2023 to February 2024. Currently, she is a Research Associate with the Vodafone Chair Mobile Communications Systems, Technische Universität Dresden. Her research interests include multi-antenna systems, massive MIMO, wireless sensor networks, spike-based communication, and machine learning.

RANJAN K. MALLIK (Fellow, IEEE) received the B.Tech. degree in electrical engineering from Indian Institute of Technology Kanpur, Kanpur, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1988 and 1992, respectively.

From August 1992 to November 1994, he was a scientist with the Defence Electronics Research Laboratory, Hyderabad, India, working on missile and EW projects. From November 1994 to January 1996, he was a faculty member of the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology Kharagpur, Kharagpur. From January 1996 to December 1998, he was a faculty member of the Department of Electronics and Communication Engineering, Indian Institute of Technology Guwahati, Guwahati. Since December 1998, he has been a faculty member of the Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, where he is currently a Professor. His research interests include diversity combining and channel modeling for wireless communications, space-time systems, cooperative communications, multiple-access systems, power line communications, molecular communications, difference equations, and linear algebra.

Dr. Mallik is a member of Eta Kappa Nu, the IEEE Communications, Information Theory, and Vehicular Technology Societies, American Mathematical Society, the International Linear Algebra Society, and the Association for Computing Machinery; a fellow of Indian National Academy of Engineering, Indian National Science Academy, The National Academy of Sciences, India, Prayagraj, Indian Academy of Sciences, Bengaluru, The World Academy of Sciences-for the advancement of science in developing countries (TWAS), The Institution of Engineering and Technology, U.K., The Institution of Electronics and Telecommunication Engineers, India, The Institution of Engineers (India) (IEI), the Asia–Pacific Artificial Intelligence Association, and the Artificial Intelligence Industry Academy; and a life member of Indian Society for Technical Education. He is a recipient of the Hari Om Ashram Prerit Dr. Vikram Sarabhai Research Award in the field of electronics, telematics, informatics, and automation, the Shanti Swarup Bhatnagar Prize in engineering sciences, the Khosla National Award, the IEI-IEEE Award for Engineering Excellence, and the J. C. Bose Fellowship. He has served as an Area Editor and an Editor for IEEE Transactions on Wireless Communications and as an Editor for IEEE Transactions on Communications.

KHALED B. LETAIEF (Fellow, IEEE) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in December 1984, August 1986, and May 1990, respectively, and the Ph.D. (Honoris Causa) degree from the University of Johannesburg, South Africa, in 2022.

He is currently an Internationally Recognized Leader of Wireless Communications and Networks. Since 1993, he has been with The Hong Kong University of Science and Technology (HKUST), where he has held many administrative positions, including Acting Provost, the Head of the Electronic and Computer Engineering Department, the Director of the Wireless IC Design Center, and the Director of Hong Kong Telecom Institute of Information Technology. While with HKUST, he was a Chair Professor and the Dean of Engineering. From September 2015 to March 2018, he joined HBKU as a Provost to help establish a research-intensive university in Qatar in partnership with strategic partners that include Northwestern University, Carnegie Mellon University, Cornell, and Texas A&M. He is also recognized by Thomson Reuters as an ISI Highly Cited Researcher and was listed among the 2020 top 30 of AI 2000 Internet of Things Most Influential Scholars.

Dr. Letaief is a member of United States National Academy of Engineering, a fellow of Hong Kong Institution of Engineers, a member of India National Academy of Sciences, and a member of Hong Kong Academy of Engineering Sciences. He was a recipient of many distinguished awards and honors, including 2007 IEEE Communications Society Joseph LoCicero Publications Exemplary Award, 2011 IEEE Communications Society Harold Sobol Award, 2016 IEEE Marconi Prize Paper Award in Wireless Communications, 2019 IEEE Communications Society and Information Theory Society Joint Paper Award, 2021 IEEE Communications Society Best Survey Paper Award, 2022 IEEE Communications Society Edwin Howard Armstrong Achievement Award, the 2024 Distinguished Purdue University Alumni Award, and over 20 IEEE Best Paper Awards. He is well recognized for his dedicated service to professional societies and IEEE, where he has served in many leadership positions. These include the Founding Editor-in-Chief of the prestigious IEEE Transactions on Wireless Communications. He also served as the President for the IEEE Communications Society (2018–2019), the world's leading organization for communications professionals with headquarter in New York City, and a member of 162 countries.