*mathematics*

MDPI

*Article*

# Dynamic Scheduling and Power Allocation with Random Arrival Rates in Dense User-Centric Scalable Cell-Free MIMO Networks

**Kyung-Ho Shin [1,2], Jin-Woo Kim [1,2], Sang-Wook Park [1,2], Ji-Hee Yu [1,2], Seong-Gyun Choi [1,2], Hyoung-Do Kim [1,2], Young-Hwan You [2,3] and Hyoung-Kyu Song [1,2,*]**

[1] Department of Information and Communication Engineering, Sejong University, Seoul 05006, Republic of Korea; shinkh1000@naver.com (K.-H.S.); kjwccm@naver.com (J.-W.K.); share1211@naver.com (S.-W.P.); wlgml5974@naver.com (J.-H.Y.); sk4753611@naver.com (S.-G.C.); gudeh8330@naver.com (H.-D.K.)
[2] Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea; yhyou@sejong.ac.kr
[3] Department of Computer Engineering, Sejong University, Seoul 05006, Republic of Korea
* Correspondence: songhk@sejong.ac.kr

**Abstract:** In this paper, we address scheduling methods for queue stabilization and appropriate power allocation techniques in downlink dense user-centric scalable cell-free multiple-input multiple-output (CF-MIMO) networks. Scheduling is performed by the central processing unit (CPU) scheduler using Lyapunov optimization for queue stabilization. In this process, the drift-plus-penalty is utilized, and the control parameter $V$ serves as the weighting factor for the penalty term. The control parameter $V$ is fixed to achieve queue stabilization. We introduce the dynamic $V$ method, which adaptively selects the control parameter $V$ considering the current queue backlog, arrival rate, and effective rate. The dynamic $V$ method allows flexible scheduling based on traffic conditions, demonstrating its advantages over fixed $V$ scheduling methods. In cases where UEs scheduled with dynamic $V$ exceed the number of antennas at the access point (AP), the semi-orthogonal user selection (SUS) algorithm is employed to reschedule UEs with favorable channel conditions and orthogonality. Dynamic $V$ shows the best queue stabilization performance across all traffic conditions. It shows a 10% degraded throughput performance compared to $V$ = 10,000. Max-min fairness (MMF), sum SE maximization, and fractional power allocation (FPA) are widely considered power allocation methods. However, the power allocation method proposed in this paper, combining FPA and queue-based FPA, achieves up to 60% better queue stabilization performance compared to MMF. It is suitable for systems requiring low latency.

**Keywords:** cell-free MIMO; scheduling; power allocation; Lyapunov optimization; control parameter $V$; SUS algorithm; queue-based FPA

**MSC:** 90B18



**Citation:** Shin, K.-H.; Kim, J.-W.; Park, S.-W.; Yu, J.-H.; Choi, S.-G.; Kim, H.-D.; You, Y.-H.; Song, H.-K. Dynamic Scheduling and Power Allocation with Random Arrival Rates in Dense User-Centric Scalable Cell-Free MIMO Networks. *Mathematics* **2024**, *12*, 1515. https://doi.org/10.3390/math12101515

Academic Editor: Sergio Gómez

Received: 4 April 2024
Revised: 3 May 2024
Accepted: 10 May 2024
Published: 13 May 2024



## 1. Introduction

The demand for wireless data has shown exponential growth over the past few decades, and it is expected to increase even further in the future. Simultaneously, user equipment (UE) demands high data rates, wide coverage, and low-latency services [1,2]. To meet these requirements, massive MIMO has been presented as a breakthrough technology in 5G [3,4]. Massive MIMO based on a large antenna array enables directional transmission to each UE, serves multiple UEs in the same cell at different locations, and leads to a significant increase in spectral efficiency (SE) performance. However, the received signal power is rapidly attenuated with the propagation distance. Therefore, even with the use of massive MIMO at the cell edge UEs may receive signals of low power and poor quality, including interference from neighboring APs located far away [5,6].

To address this, the concept of a CF-MIMO network has emerged. A CF-MIMO network is a network where geographically distributed APs jointly serve UEs, and APs are connected to the CPU by front-haul links. The advantage of a CF-MIMO network is that it reduces the propagation distance between APs and UEs, ensuring good signal quality regardless of UE location [6–9].

Serving all UEs by every AP in the CF network is not scalable. If a distant AP serves a UE, it uses wireless resources, but it does not contribute to the useful signal power of the UE. Therefore, it is essential to perform clustering in a practical way to allow limited APs to serve UEs [10,11]. Among the CF clustering options, scalable user-centric clustering is widely adopted. This approach conducts clustering based on the channel gain between APs and UEs, ensuring that UEs receive signals of good quality only from serving APs.

This paper focuses on the scenario where UEs outnumber antennas, unlike most papers that consider more antennas than UEs. In practice, due to the installation cost constraints of APs, a CF-MIMO network that exceeds the number of antennas is challenging. Additionally, it is expected that the number of UEs will grow at a faster rate than the number of antennas in the future [12,13]. Consequently, not all users can be scheduled in the same time slot.

Resource allocation, such as scheduling and power allocation, is crucial in wireless communication due to random or bursty traffic. Furthermore, it is anticipated that the demand for low latency will increase in 6G networks. In this paper, we implement scheduling that satisfies low latency while maximizing the throughput using Lyapunov optimization in the CF-MIMO network, providing an environment where UEs can achieve consistent quality performance regardless of their location. Lyapunov optimization is employed to stabilize the queue in queueing networks and minimize the penalty function. The penalty function is associated with network utility, such as time-average power, packet drop, and throughput. In this paper, the penalty function is configured with a negative value for throughput, creating a trade-off relationship with queue stabilization. The non-negative control parameter $V$ serves as the weight of the penalty function. The value of $V$ can be chosen to influence the desired performance. Since the parameter $V$ is fixed in most papers, scheduling is conducted regardless of the current queue state and traffic conditions. However, in this paper, we provide a flexible scheduling approach by adjusting the value of the parameter $V$ for each time slot.

### *1.1. Related Works*

There has been some research on Lyapunov optimization [13–18]. The authors of [14] provided a comprehensive methodology for Lyapunov drift-plus-penalty optimization, aiming to stabilize the queue while minimizing the penalty function defined in the system. The authors of [15] established a trade-off relationship between queue stability and throughput and employed Lyapunov optimization for resource allocation in LTE-based systems. Additionally, they addressed the maximum weighted sum-rate power allocation problem in the physical layer using a branch-and-bound algorithm. However, the approach utilizes a scheduling algorithm using a fixed parameter $V$ and a power allocation method that has the drawback of high complexity due to iterative power allocation optimization. The authors of [16] focused on stabilizing virtual queues through Lyapunov optimization in a cell-free network and used a simple block-coordinate descent algorithm to converge to the local optimal for power allocation, similar to [15]. The approach restricts the scenario to cases where the arrival rate is smaller than the generated data rates. Its scheduling technique does not consider scenarios with high arrival rates. The authors of [17] employed Lyapunov optimization for delay control by setting the penalty function as the negative value of the reward for format selection. They introduced a novel approach that separates the terms for format selection and transmission scheduling decisions, allowing the optimization to proceed independently for the two decisions. The approach establishes a trade-off relationship between information quality and scheduling. However, it is not suitable for systems aiming to maximize data rates. The authors of [18] conducted scheduling to prevent packet drops by configuring the $V$ value for each time slot. They addressed environments where the queue's capacity is

limited or a hard delay constraint exists and considered both single-queue and multi-queue models. However, the determination of the optimal value of the parameter $V$ for the system is challenging due to the cost of quantized packet drops.

Scheduling problems in CF networks have been discussed in previous papers [12,13,19,20], which conducted scheduling considering throughput fairness for each time slot in environments where the number of antennas is similar to the number of UEs. The authors of [13] considered active UEs to achieve maximum system performance. Similarly, in this paper, we adopt active UEs that can lead to maximum performance in a dense user-centric scalable cell-free environment. This paper also utilizes a fixed parameter $V$ to derive the results. The authors of [12,19,20] optimized scheduling, power allocation, and beamforming in a cell-free environment. In addition, the authors of [19] utilized the signal-to-leakage-plus-interference-plus-noise ratio (SLINR) metric for optimization in a multiple-CU environment and provided three methods for calculating leakage values. The authors of [20] proposed a method for solving the resource allocation optimization problem in a massive MIMO cell-free system using semidefinite relaxation (SDR) and sequential convex approximation (SCA) approaches within polynomial time. Since the authors of [12,19,20] did not consider scheduling with the queue backlog, applying their approaches to delay-sensitive systems may pose challenges.

The authors of [21] optimized the receive beamforming and the reflecting coefficient matrix in an active RIS (Reconfigurable Intelligent Surface) to maximize the received signal power. They also suggested the number of reflecting elements that amplify the signal. The approach showed superior achievable rates compared to passive RIS. The authors of [22] optimized the pilot length and pilot transmission power to maximize data error probability and throughput under limited power. They solved the problem using truncated channel inversion and an iterative optimization algorithm. The authors of [23] discussed the utilization of MEC (mobile edge computing) in latency-sensitive systems within CF massive MIMO networks. They presented the SECP (successful edge computing probability) for target latency and examined the impact of the SECP and AP density on energy efficiency. The authors of [24] introduced a method for estimating 2D DOA in mmWave polarized massive MIMO systems using compressive sampling. They proposed an approach suitable for real-time implementations due to its low complexity. The authors of [25] optimized transmission power and NOMA pairing using the A3C (asynchronous advantage actor-critic)-based ESSO (energy-efficiency secure offloading) scheme, a deep reinforcement learning-based solution. The approach resulted in improved energy consumption and average secrecy probability. However, the above-mentioned papers did not consider scheduling to reduce latency.

### *1.2. Contributions*

In this paper, the number of UEs clustered on an AP is less than the number of antennas on the AP, making it impossible to serve them simultaneously in a time slot. Therefore, active UEs should be scheduled in each time slot.

First, in this paper, in a dense user-centric scalable CF MIMO network, dynamic scheduling is performed on the CPU based on the queue backlog and instantaneous channel state information (CSI) of each user using the drift-plus-penalty method. During scheduling, the number of active UEs that can achieve maximum system performance is selected. Additionally, the value of the parameter $V$ is adaptively set based on the queue backlog of each UE to efficiently handle bursty traffic. The proposed method is analyzed by comparing it with the scheduling method using a fixed parameter $V$.

Since the scheduling is performed on the CPU using the drift-plus-penalty method, the number of UEs scheduled to APs can exceed the number of antennas on each AP. Therefore, the CPU utilizes the SUS algorithm for APs to reschedule UEs when the number of scheduled UEs exceeds the number of antennas.

Furthermore, the local maximum ratio (MR) precoder is simply designed at each AP to verify the performance. In distributed operations, APs locally perform power allocation. We introduce two power allocation methods: FPA is suitable for all traffic scenarios and

has outstanding throughput performance, and queue-based FPA is capable of satisfying low-latency requirements in low-traffic situations. Therefore, we combine FPA with queue-based FPA to ensure proper operation across all traffic intervals. We provide simulation results of these two methods compared with the computationally complex MMF and sum SE maximization.

### *1.3. Paper Structure and Notations*

The rest of this paper is organized as follows. Section 2 describes the downlink CF-MIMO network model and the K-queue network model. In Section 3, we present the CPU scheduling method using Lyapunov optimization and the SUS algorithm. Then, the heuristic FPA and queue-based FPA power allocation methods are proposed. Section 4 evaluates the simulation results. Finally, the conclusions of this paper are presented in Section 5.

In this paper, vectors and matrices are denoted with bold letters. $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $(\cdot)^{-1}$, and $\|\cdot\|$ denote the transpose, conjugate, conjugate transpose, inverse, and norm, respectively. $\mathbb{E}\{\cdot\}$ denotes the expectation value, and $\mathcal{CN}(\cdot,\cdot)$ denotes the circularly symmetric complex Gaussian distribution. $\mathbf{0}$ and $\mathbf{I}$ represent an all-zero matrix and an identity matrix, respectively.

## 2. System Model

### *2.1. The Network Model*

Figure 1 illustrates a dense user-centric CF MIMO network. There are $L$ APs equipped with $M$ antennas, and all APs are connected to a single CPU. The network includes $K$ UEs with a single antenna. It is assumed that the number of UEs exceeds the available resources in a single time slot. Each UE forms a cluster based on the channel gains from surrounding APs, denoted as $\mathbf{C}_k$ for UE $k$. The AP $l$ cluster is denoted as $\mathbf{A}_l$. Each UE should establish connections with at least one AP when a cluster is formed. Therefore, each UE always includes the AP with the highest channel gain in the cluster, and APs with channel gains exceeding the threshold $\gamma$ in the set of APs $\{1, ..., L\}$ are also included in $\mathbf{C}_k$. This process is mathematically expressed as follows:

$$\mathbf{C}_k = \{l : \beta_{lk} \geq \gamma, l \in \{1, ..., L\}\} \cup \{\arg\max_l \beta_{lk}\}, \tag{1}$$

where $\beta_{lk}$ represents the average of the large-scale fading coefficient between AP $l$ and UE $k$, expressed as path loss and shadow fading. We use a block fading model to denote the coherence interval $\tau_c$, during which the channel remains unchanged. Additionally, each AP configures a cluster such that the pilot length $\tau_p$ is equal to the number of UEs in the cluster to mitigate pilot contamination effects.

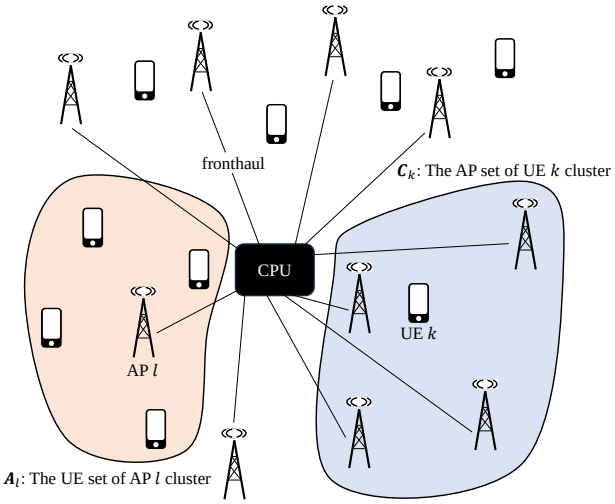


**Figure 1.** Example of user-centric cell-free MIMO system model. The red area and blue area represent the UE set of AP $l$ cluster and the AP set of UE $k$ cluster, respectively.

The K-queue network model considered in this paper is depicted in Figure 2. Each UE has its own distinct queue. At time slot t, the queue backlog vector is represented as $\mathbf{Q}(t) = [Q_1(t), Q_2(t), ..., Q_K(t)]$. $Q_k(t)$ denotes the queue backlog for UE $k$ at time slot $t$ subject to $Q_k(t) \geq 0$. Additionally, $\mathbf{a}(t) = [a_1(t), a_2(t), ..., a_K(t)]$ represents the UE's arrival rate vector, and $\mathbf{r}(t) = [r_1(t), r_2(t), ..., r_K(t)]$ represents the departure rate. In this paper, the arrival rate is assumed to be an uncontrollable random event. $\mathbf{r}(t)$ is determined through scheduling. The CPU estimates the effective rate using the given CSI, and the effective rate can be obtained through numerous realizations using instantaneous mutual information.

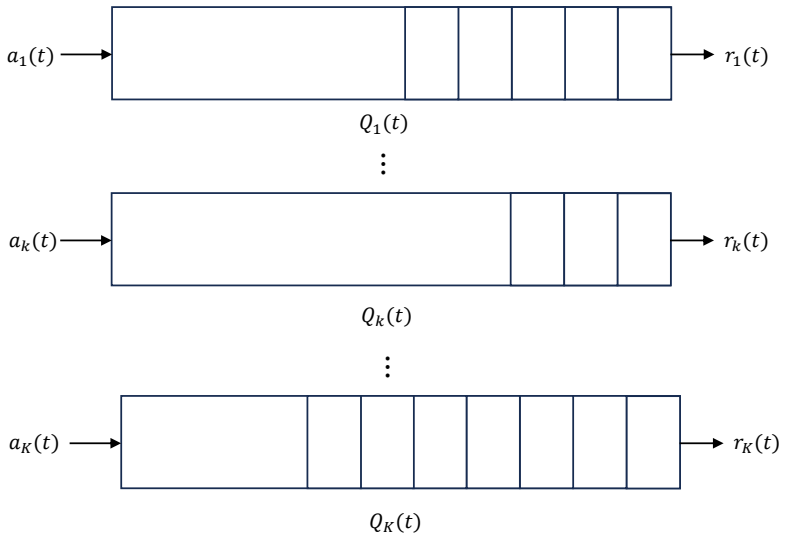


**Figure 2.** The K-queue network model on the CPU.

The channel between AP $l$ and UE $k$ is denoted as $\mathbf{h}_{lk} \in \mathbb{C}^{M\times 1}$, and it is expressed as follows:

$$\mathbf{h}_{lk} = \mathbf{R}_{lk}^{1/2}\mathbf{g}_{lk}, \tag{2}$$

where $\mathbf{R}_{lk} \in \mathbb{C}^{M\times M}$ is the large-scale fading diagonal matrix and $\mathbf{g}_{lk} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_M)$ is the small-scale fading vector between AP $l$ and UE $k$.

In the downlink, the signal received by UE $k$ at time slot $t$ can be expressed as follows:

$$y_k(t) = \underbrace{\sum_{l\in\mathbf{C}_k} s_{lk}(t)\mathbf{h}_{lk}^H\mathbf{w}_{lk}x_k}_{\mathrm{DS}_k} + \underbrace{\sum_{k'\in\{1,...,K\},k'\neq k}\sum_{l'\in\mathbf{C}_{k'}} s_{l'k'}(t)\mathbf{h}_{l'k}^H\mathbf{w}_{l'k'}x_{k'}}_{\mathrm{IS}_k} + n_k. \tag{3}$$

$\mathrm{DS}_k$ and $\mathrm{IS}_k$ denote the desired signal and interference signal, respectively. $s_{lk}(t)$ is 1 if UE $k$ is scheduled at AP $l$ at time slot $t$; otherwise, it is 0. $\mathbf{w}_{lk}$ is the precoder vector for UE $k$ of AP $l$. $x_k$ is the UE's transmission symbol with an average of 0 and a variance of 1. The second term is the interference signal of UE $k$, which is not the signal intended for UE $k$. $n_k \sim \mathcal{CN}(0, \sigma^2)$ is the additional white Gaussian noise (AWGN) of UE $k$.

### *2.2. Channel Estimation*

It is assumed that geographically distributed APs locally estimate channels. The estimated channel information is shared with the CPU through a front-haul link, and it is crucial for scheduling by the CPU. It is used by the APs for precoder design and power allocation. The UEs in $\mathbf{A}_l$ are assigned different pilot signals with a length of $\tau_p$ to minimize pilot contamination during channel estimation. Each pilot signal is represented as $\phi_1, ..., \phi_{\tau_p}$ and is orthogonal to the others. Pilot signals are designed to satisfy $\phi_t \in \mathbb{C}^{\tau_p\times 1}$ for $t = 1, ..., \tau_p$. Since UEs within $\mathbf{A}_l$ use orthogonal pilot signals, there is no pilot contami-

nation within the cluster. However, UEs in other AP clusters using the same pilot sequence can lead to pilot contamination. The set of UEs sharing pilots with UE $k$ is denoted as $\mathcal{P}_k$. The pilot signals received by AP $l$ during the channel estimation phase can be expressed as follows:

$$\mathbf{y}_l^{pilot} = \sum_{i=1}^{K} \sqrt{p_i}\mathbf{h}_{li}\phi_{t_i}^T + \mathbf{N}_l. \tag{4}$$

In the channel estimation phase, all UEs in the network transmit pilot signals with the same power. $t_i$ represents the pilot signal index of UE $i$, and $\mathbf{N}_l \in \mathbb{C}^{M\times\tau_p}$ denotes the noise component entering the receiving antennas.

To estimate the UE $k$ channel at AP $l$, the conjugate transpose vector of $\phi_{t_k}^T$ is multiplied to eliminate other pilot components. This can be represented by the following equation:

$$\widehat{\mathbf{h}}_{lt_k}^{ls} = \sum_{i=1}^{K} \frac{\sqrt{p_i}}{\sqrt{\tau_p}}\mathbf{h}_{li}\phi_{t_i}^T\phi_{t_k}^* + \frac{1}{\sqrt{\tau_p}}\mathbf{N}_l\phi_{t_k}^*, \tag{5}$$

where if $i$ and $k$ are equal, $\phi_{t_k}^T\phi_{t_k}^* = \tau_p$; otherwise, it is 0. And $\phi_{t_i}^*$ represents the conjugate version of $\phi_{t_i}$.

To suppress the interference from the shared pilot and noise components, it is multiplied by the $\Psi_{t_kl}^{-1}$ term, which is the sum of the large-scale fading coefficient for the shared pilot and the noise. After that, it is multiplied again by the large-scale fading coefficient of UE $k$ to perform minimum mean square error (MMSE) channel estimation. The MMSE channel estimation expression is as follows:

$$\widehat{\mathbf{h}}_{lk} = \sqrt{p_k\tau_p}\mathbf{R}_{lk}\Psi_{lt_k}^{-1}\widehat{\mathbf{h}}_{lt_k}^{ls} = \sqrt{p_k\tau_p}\mathbf{R}_{lk}\left(\sum_{i\in\mathcal{P}_k} p_i\tau_p\mathbf{R}_{li} + \sigma^2\mathbf{I}_M\right)^{-1}\widehat{\mathbf{h}}_{lt_k}^{ls}, \tag{6}$$

where $\widehat{\mathbf{h}}_{lk}$ is the estimated channel of UE $k$ at AP $l$. Additionally, $\sigma^2$ is the power value of the term obtained by multiplying $\frac{\phi_{t_i}^*}{\sqrt{\tau_p}}$ with the receiver noise $\mathbf{N}_l$.

The MMSE channel estimated locally at the AP is transmitted to the CPU via front-haul links. Then, the CPU performs scheduling using the estimated channel and the queue backlog of each UE.

## 3. The Proposed Methods

In this section, we propose a scheduling method on the CPU that ensures queue stabilization regardless of traffic conditions, utilizing the variable parameter $V$. Additionally, we propose a power allocation method that satisfies low-latency requirements under low-traffic conditions.

### *3.1. Proposed Scheduling on the CPU*

#### 3.1.1. Lyapunov Optimization

Figure 3 illustrates the scheduling process of the CPU scheduler. At time slot $t$, if $\widehat{\mathbf{H}}(t)$, estimated at the APs, and $\mathbf{Q}(t)$ are utilized, $\frac{LM}{2}$ UEs are scheduled in the Lyapunov optimization phase, and this set of UEs is denoted as $\mathbf{K}_{act,LO}$. If $\frac{LM}{2}$ UEs are served, it can maximize network throughput performance [13]. Among $\mathbf{K}_{act,LO}$, the set of UEs that the AP needs to serve is denoted as $\mathbf{s}_l^{'}$, and it can be obtained as $\mathbf{A}_l \cap \mathbf{K}_{act,LO}$. If $|\mathbf{s}_l^{'}| > M$ for AP $l$, the SUS algorithm is applied for rescheduling. The finally scheduled UEs by AP $l$ are represented as $\mathbf{s}_l$, and the scheduled UEs in the network are denoted as $\mathbf{K}_{act} = \bigcup_{l=1}^{L}\mathbf{s}_l$. The CPU determines the effective data rate with the estimated CSI. First, the CPU computes the mutual information for each UE as follows:

$$I_k(t) = log\left(1 + \frac{\|\sum_{l\in\mathbf{C}_k}\hat{\mathbf{h}}_{lk}^H\mathbf{w}_{lk}\|^2}{\sum_{k'\in\{1,\ldots,K\},k'\neq k}\|\sum_{l'\in\mathbf{C}_{k'}}\hat{\mathbf{h}}_{l'k}^H\mathbf{w}_{l'k'}\|^2 + \sigma^2}\right). \tag{7}$$

Each UE has mutual information for the most recent 1000 time slots. The CPU utilizes it to compute the allocated rate to each UE. The allocated rate is calculated using the following formula [13]:

$$R_k = \arg\max_{\mathcal{R}} \mathcal{R} \times \mathbb{P}(\mathcal{R} < I_k(t)), \tag{8}$$

where $\mathbb{P}(\mathcal{R} < I_k(t))$ is the probability of an event $\mathcal{R} < I_k(t)$. Then, we use the allocated rate to determine that the effective data rate is given as $r_k^*(t) = R_k \times 1\{R_k < I_k(t)\}$. $1\{R_k < I_k(t)\}$ is the indicator function of an event $R_k < I_k(t)$.

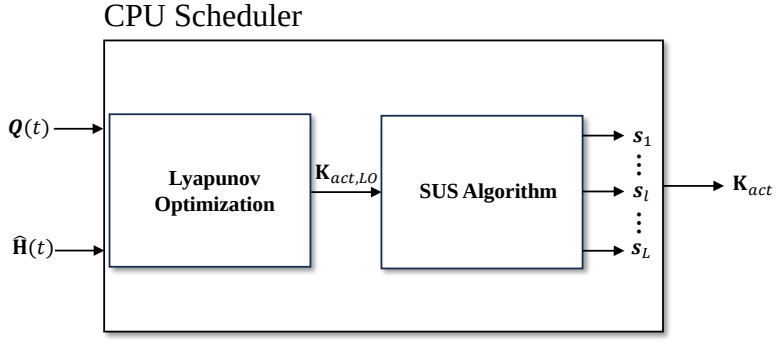


**Figure 3.** Diagram of the CPU scheduler.

The goal is to maximize the effective rate while ensuring queue stability. To achieve this, the queue for UE $k$ in the next time slot $t+1$ is updated. The prediction of $Q_k(t+1)$ can be expressed by the arrival rate, effective rate, and $Q_k(t)$ as follows:

$$Q_{k,pred}(t+1) = \max[Q_k(t) - r_k^*(t), 0] + a_k(t). \tag{9}$$

Then, the quadratic Lyapunov function with $Q_k(t)$ can be formulated as follows:

$$\mathcal{L}(\Theta(t)) = \frac{1}{2}\sum_{k=1}^{K} Q_k^2(t). \tag{10}$$

We can easily determine the congestion level of the network using the quadratic Lyapunov function. The one-step conditional Lyapunov drift is a measure that quantifies the difference in congestion levels between two time slots as a scalar value. Minimizing this drift is equivalent to stabilizing the queue. The following equation represents the one-step conditional Lyapunov drift [14,16,17].

$$\begin{aligned} \Delta(\Theta(t)) &= \mathbb{E}\Big[\mathcal{L}(\Theta_{pred}(t+1)) - \mathcal{L}(\Theta(t))\Big] \\ &= \frac{1}{2}\sum_{k=1}^{K} \mathbb{E}\Big[Q_{k,pred}^2(t+1) - Q_k^2(t)\Big]. \end{aligned} \tag{11}$$

Therefore we can obtain the inequality expression for $Q_{k,pred}^2(t+1)$ [16]:

$$\begin{aligned} Q_{k,pred}^2(t+1) &\le [Q_k(t) - r_k^*(t)]^2 + {a_k}^2(t) + 2a_k \max[Q_k(t) - r_k^*(t), 0] \\ &\le Q_k^2(t) + \{r_k^*(t)\}^2 + {a_k}^2(t) - 2Q_k(t)[r_k^*(t) - a_k(t)]. \end{aligned} \tag{12}$$

From (11), the following equation is obtained:

$$\begin{aligned}\Delta(\Theta(t)) &\leq \frac{1}{2}\sum_{k=1}^{K}\mathbb{E}\Big[\{r_k^*(t)\}^2 + {a_k}^2(t)\Big] - \sum_{k=1}^{K}Q_k(t)[r_k^*(t) - a_k(t)] \\ &\leq B - \sum_{k=1}^{K}Q_k(t)[r_k^*(t) - a_k(t)],\end{aligned} \tag{13}$$

where $B = \frac{1}{2}\sum_{k=1}^{K}\left[r_{max}^2 + a_{max}^2\right]$ is constant. $r_{max}$ is the maximum rate that UE $k$ can achieve, and $a_{max}$ is the maximum arrival rate in the queue at once in a time slot. A drift-plus-penalty equation is obtained by adding a penalty term $-V(t)\sum_{k=1}^{K} log(1 + r_k^*(t))$ to the drift term above. By minimizing this equation, scheduling is performed in order to stabilize the queue and maximize the throughput. The drift-plus-penalty equation is expressed as follows:

$$\begin{aligned}&\Delta(\Theta(t)) - V(t)\sum_{k=1}^{K} log(1 + r_k^*(t)) \\ &\qquad \leq B - \sum_{k=1}^{K}Q_k(t)[r_k^*(t) - a_k(t)] - V(t)\sum_{k=1}^{K} log(1 + r_k^*(t)),\end{aligned} \tag{14}$$

where $B$ is a constant and $Q_k(t)$, $r_k^*(t)$, and $a_k(t)$ represent the queue backlog, the effective rate, and the arrival rate of UE $k$, respectively.

We propose an adaptive approach in which the control parameter $V$ is dynamically adjusted. This method is called dynamic $V$ in this paper. The parameter $V(t)$ compromises between throughput and queue stabilization. The scheduling strategy can be flexibly changed by varying $V(t)$ over time. $V(t)$ is determined based on the constrained queue size $Q_{max}$ and $V_{max}$. $Q_{max}$ and $V_{max}$ are suitably chosen in the CPU. In particular, $Q_{max}$ is related to the delay allowed by the system. If the arrival rate is greater than the effective data rate, $V(t) = V_{max}$ is the fastest strategy to clear the queue backlog. However, if the arrival rate is less than the effective data rate, the closer $\max[\mathbf{Q}(t)]$ is to $Q_{max}$, the closer $V(t)$ is to 0, stabilizing the queue and preventing data drop.

$$V(t) = \begin{cases} V_{max}\frac{Q_{max} - \max[\mathbf{Q}(t)]}{Q_{max}}, & \mathbb{E}[\mathbf{r}^*(t)] > \mathbb{E}[\mathbf{a}(t)] \\ V_{max}, & \text{Otherwise.} \end{cases} \tag{15}$$

We proceed with scheduling to minimize the upper bound of the drift-plus-penalty in (14). Since $B$ is a constant and $\mathbf{a}(t)$ is a random event, they are excluded from the scheduling process. The scheduling process of $\mathbf{K}_{act,LO}$ can be expressed by the following equation:

$$\max_{s_k(t)} \quad \sum_{k=1}^{K}Q_k(t)r_k^*(t) + V(t)\sum_{k=1}^{K} log(1 + r_k^*(t)) \quad \forall t. \tag{16}$$

3.1.2. SUS Algorithm

In (16), the set $\mathbf{K}_{act,LO}$ scheduled by the CPU during the Lyapunov optimization phase needs to be served by distributed APs. However, when serving from AP $l$, $|\mathbf{s}_l^{'}(t)|$ may exceed the number of antennas at the AP $l$. Before employing the SUS algorithm, the scheduler using Lyapunov optimization does not consider the clustering status of APs and schedules $\frac{LM}{2}$ UEs. To address this, the CPU uses the SUS algorithm to reschedule UEs that exceed the number of antennas at AP $l$. The SUS algorithm selects UEs that ensure good channel quality. When these UEs are scheduled and served simultaneously, they are semiorthogonal to each other, resulting in significantly improved throughput performance when using the MR precoder. This maintains queue stability better and improves throughput performance compared to randomly selecting UEs for scheduling at APs. The SUS algorithm operating on the CPU is described in Algorithm 1 [26,27].

If scheduling is performed, $Q_k(t+1)$ can be expressed as follows:

$$Q_k(t+1) = \max[Q_k(t) - r_k(t), 0] + a_k(t). \tag{17}$$

**Algorithm 1:** SUS algorithm in the CPU

**Input:** $\widehat{\mathbf{h}}(t); \mathbf{s}_l^{'}(t);$
**Output:** The set of scheduled UEs on all APs $\mathbf{s}_l(t);$
**Initialization:** $l \leftarrow 1; i \leftarrow 1;$
**while** $l < L$ **do**
  **if** $M < |\mathbf{s}_l^{'}(t)|$ **then**
    **while** $i < M$ **do**
      **foreach** $k$ ***in*** $\mathbf{s}_l^{'}(t)$ **do**
        $\mathbf{q}_k^{'} = \widehat{\mathbf{h}}_k(t) - \sum_{j=1}^{i-1} \frac{\widehat{\mathbf{h}}_k(t)\mathbf{q}_j^H \mathbf{q}_j}{\|\mathbf{q}_j\|_2};$
      **end**
      $i_{opt} = \arg\max_{k \in \mathbf{s}_l^{'}} \|\mathbf{q}_k^{'}\|_2;$
      $\mathbf{s}_l \leftarrow \mathbf{s}_l \cup \{i_{opt}\};$
      $\mathbf{s}_l^{'} \leftarrow \mathbf{s}_l^{'} \backslash \mathbf{s}_l;$
      $\mathbf{q}_i = \mathbf{q}_{i_{opt}}^{'};$
      $i \leftarrow i + 1;$
    **end**
  **else**
    $\mathbf{s}_l(t) \leftarrow \mathbf{s}_l^{'}(t);$
  **end**
**end**

### *3.2. Proposed Power Allocation*

Power allocation is also carried out locally at each AP. Instead of computationally expensive methods such as sum SE maximization or max-min fairness, a heuristic power allocation approach known as FPA is adopted by the APs [6]. The power allocation process using FPA at each AP can be expressed as follows:

$$\rho_{kl} = \begin{cases} \rho_{max} \frac{(\beta_{kl})^{\nu}}{\sum_{i \in \mathbf{s}_l} (\beta_{il})^{\nu}}, & k \in \mathbf{s}_l \\ 0, & k \notin \mathbf{s}_l. \end{cases} \tag{18}$$

The range of $\nu$ is $-1 \leq \nu \leq 1$. As $\nu$ approaches $-1$, it considers fairness among UEs similar to max-min fairness. When power is allocated, it allocates low power to UEs in good channel conditions. Therefore, it does not effectively utilize the network's capacity. In environments with high traffic, this power allocation method compromises queue stability. As $\nu$ approaches 1, it considers network capacity similar to SE maximization. It allocates high power to UEs in good channel conditions and low power to others. This approach sacrifices UE fairness to some extent but shows proper queue stabilization performance in high-traffic scenarios. In scenarios with low traffic, FPA with $\nu$ = 0.5 may exhibit adequate queue stabilization performance, but there is a possibility that it cannot meet the requirements of systems demanding low latency. Therefore, in low-traffic scenarios, the queue-based FPA proposed in this paper is applied, considering the current queue backlog and effective rate. It can be expressed by the following equation:

$$\rho_{kl} = \begin{cases} \rho_{max} \frac{(Q_k(t) r_k^*(t))^{\nu}}{\sum_{i \in \mathbf{s}_l} ((Q_i(t) r_i^*(t))^{\nu}}, & k \in \mathbf{s}_l \\ 0, & k \notin \mathbf{s}_l. \end{cases} \tag{19}$$

Queue-based FPA demonstrates high queue stabilization performance in low-traffic scenarios by utilizing only the information used for scheduling. It maintains low computational complexity in power allocation. Therefore, FPA with $\nu = 0.5$ is employed when $\mathbb{E}[\mathbf{r}^*(t)] < \mathbb{E}[\mathbf{a}(t)]$. Otherwise, queue-based FPA with $\nu = 0.5$ is applied. In this paper, this power allocation method is called FPA + queue-based FPA with $\nu = 0.5$. Section 4 compares the performance of FPA + queue-based FPA, FPA alone, MMF, and sum SE maximization.

## 4. Performance Evaluation

In this section, we present simulation results based on the power allocation method described in Section 3.2. Then, we compare the scheduling strategy that uses a fixed $V$ value with the dynamic $V$ approach proposed in this paper. The conventional approach in the Lyapunov optimization phase employs a fixed scheduling strategy with a constant $V$ value. However, this method lacks flexibility, as the fixed $V$ value does not adapt to changes in traffic and does not consider the current queue backlog. Therefore, the proposed dynamic $V$ approach takes into account the queue backlog, arrival rate, and effective rate for adaptive scheduling. We also briefly compare the SUS algorithm with random selection in Section 4.3.

The simulation setup and conditions were as follows:

- We established a dense user-centric scalable CF-MIMO network with $L = 45$ APs, $M = 4$ antennas per AP, and $K = 200$ UEs with a single antenna. Additionally, the following parameters were assumed: a coherence interval symbol length of $\tau_c = 200$, and a pilot signal length of $\tau_p = 10$. Each UE transmitted pilot signals at a power of $p_i = 100$ mW, and the downlink maximum power of APs was assumed to be $\rho_{max} = 200$ mW.
- When Lyapunov optimization was applied in the CPU scheduler, $|\mathbf{K}_{act,LO}|$ was configured to $\frac{LM}{2}$ to maximize network capacity utilization.
- The arrival rate $a_k(t)$ was set as $Bernoulli(p_k) \times A_{max}$, where $p_k = 0.3$, and every 4000 time slots, $A_{max}$ changed to $4\tau_c$, $3\tau_c$, $2\tau_c$, $1.5\tau_c$, and $1\tau_c$, gradually reducing the arrival rate. This meant that the traffic changed every 4000 time slots. It allowed us to verify how each scheduling algorithm and power allocation method operated in all traffic intervals. Assuming $V_{max} = 10{,}000$ and $Q_{max} = 100{,}000$, each scheduling algorithm was simulated while varying traffic over 20,000 slots to obtain the simulation results.
- Based on the estimated channels, precoding was performed at the APs. In this paper, a simple MR precoder was adopted to reduce the burden of APs. AP $l$ simply obtained the precoder of UE $k$ as follows:

$$\mathbf{w}_{kl} = \sqrt{\frac{\rho_{lk}}{\mathbb{E}\{\|\widehat{\mathbf{h}}_{lk}\|^2\}}}\widehat{\mathbf{h}}_{lk} \tag{20}$$

### 4.1. Power Allocation Comparison

Figures 4 and 5 compare the performance of queue stabilization across time slots and the cumulative distribution function (CDF) of the throughput performance for various power allocation methods.

In Figure 4, the MMF allocation, optimizing user fairness from the SE perspective, is more effective for queue elimination in low-traffic intervals than in high-traffic intervals. Sum SE maximization aims to maximize network SE and exhibits the best performance in high-traffic intervals but does not guarantee sufficient performance in low-traffic intervals. When FPA with $\nu = 0.5$ is utilized, it provides a simple power allocation that ensures appropriate performance for queue stabilization. FPA + queue-based FPA with $\nu = 0.5$ utilizes FPA with $\nu = 0.5$ in high-traffic scenarios and switches to queue-based FPA in low-traffic scenarios. It enters a low-traffic period starting from time slot 8000. Then, queue-based FPA, which allocates more power to UEs with a high queue backlog, is applied. Consequently, it removes the queue backlog faster than FPA with v = 0.5. In low-traffic

intervals, it outperforms MMF by exhibiting lower queue backlogs, making it a viable power allocation method in systems where low latency is crucial.

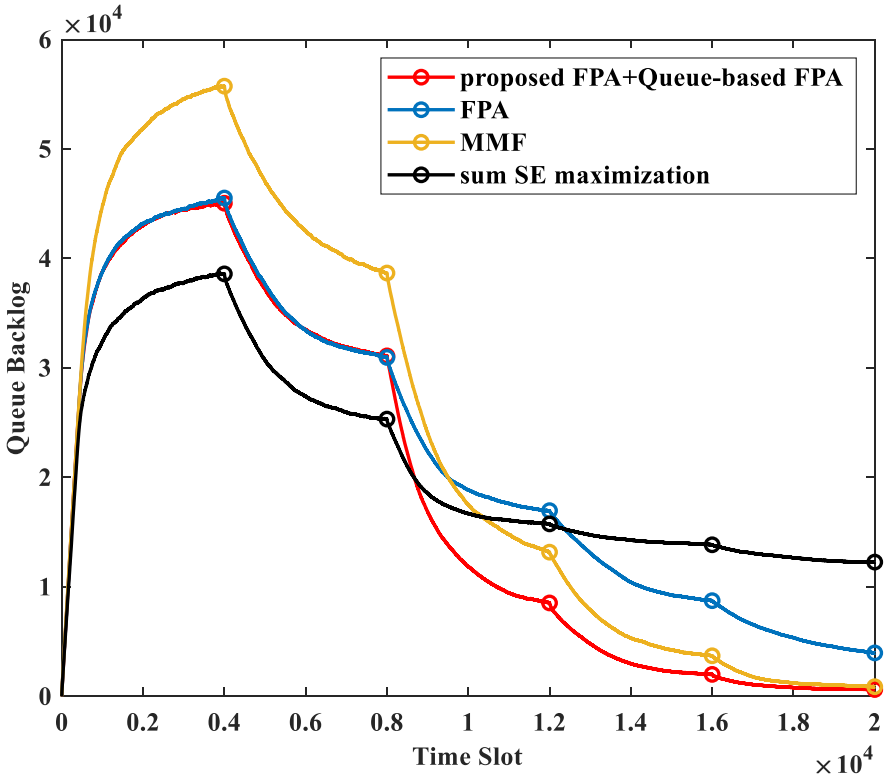


**Figure 4.** Queue backlog comparison across time slots for FPA + queue-based FPA with $\nu = 0.5$, FPA with $\nu = 0.5$. MMF. and sum SE maximization.

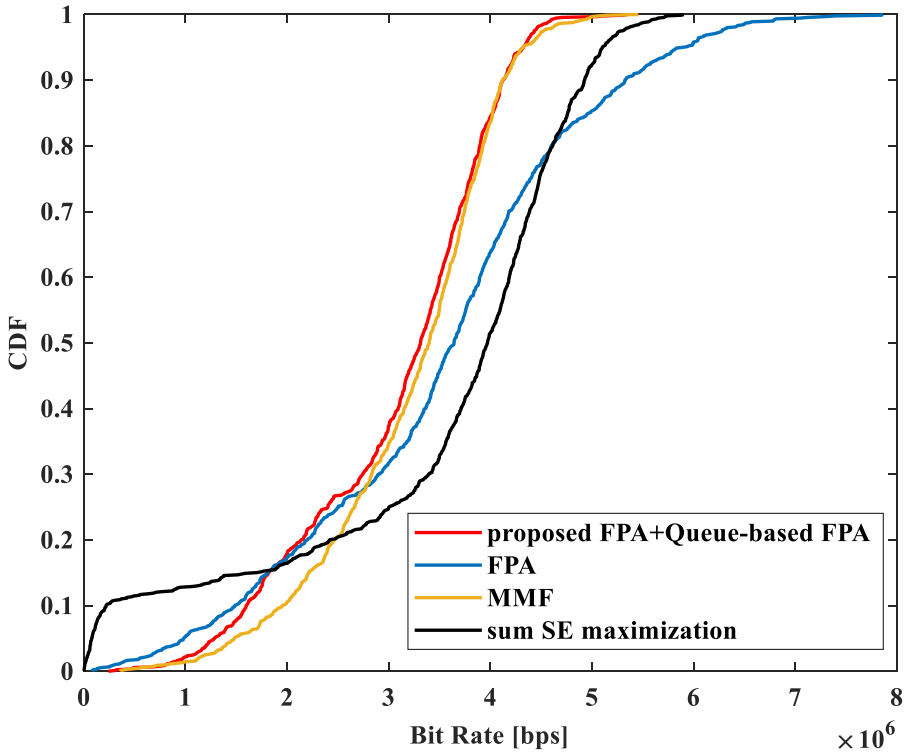


**Figure 5.** CDF of throughput comparison for FPA + queue-based FPA with $\nu = 0.5$, FPA with $\nu = 0.5$, MMF, and sum SE maximization.

Figure 5 shows that MMF exhibits good user fairness performance but low throughput performance. However, due to good user fairness, the variation in throughput among UEs is not significant. The sum SE maximization method exhibits the best performance for intermediate UEs, with throughput performance ranging from 20% to 80%. FPA with $\nu = 0.5$ also demonstrates appropriate throughput performance compared to other power allocations. The 95%-likely throughput performance is 6.2 times greater than that of sum SE maximization. The throughput performance of FPA + queue-based FPA with $\nu = 0.5$ is similar to that of MMF because it considers queue stabilization power allocation in low-traffic intervals.

### *4.2. Comparison of Scheduling Algorithms*

We compare the performance of scheduling algorithms using dynamic $V$ and fixed $V$, and we also evaluate a modified scheduling method where UEs with $r_k^*(t) > Q_k(t)$ are not scheduled. If a UE with $r_k^*(t) > Q_k(t)$ is scheduled, it may incur a loss in system throughput. Therefore, we combine it with scheduling considering throughput $V = 10{,}000$ and perform a performance comparison. This scheduling method is represented as modified $V = 10{,}000$ in the performance graph. Figures 6 and 7 illustrate this comparison using FPA with $\nu = 0.5$. The fixed $V$ approach is compared with $V = 1$ and $V = 10{,}000$.

Figure 6 compares the queue stabilization performance of the four scheduling methods. When traffic is high, the performance of dynamic $V$ is similar to that of $V = 10{,}000$. However, in the case of modified $V = 10{,}000$, UEs with an effective rate larger than the queue backlog are not scheduled, leading to more efficient removal of the system queue backlog. When traffic is low, the performance of dynamic $V$ is similar to that of $V = 1$, indicating that the proposed dynamic $V$ approach flexibly performs queue stabilization scheduling depending on traffic conditions. However, modified $V = 10{,}000$ exhibits lower performance in low-traffic scenarios. In low-traffic situations, there are fewer UEs with queue backlogs larger than their effective rates, reducing the ability to remove the queue backlog.

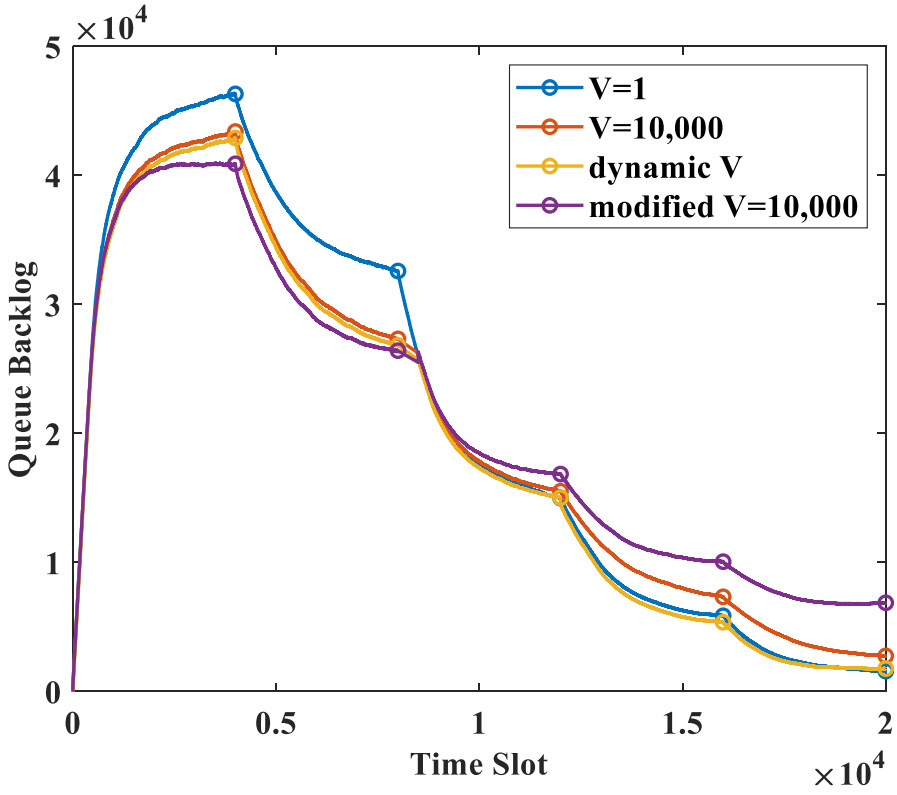


**Figure 6.** Queue backlog with varying control parameter $V$.

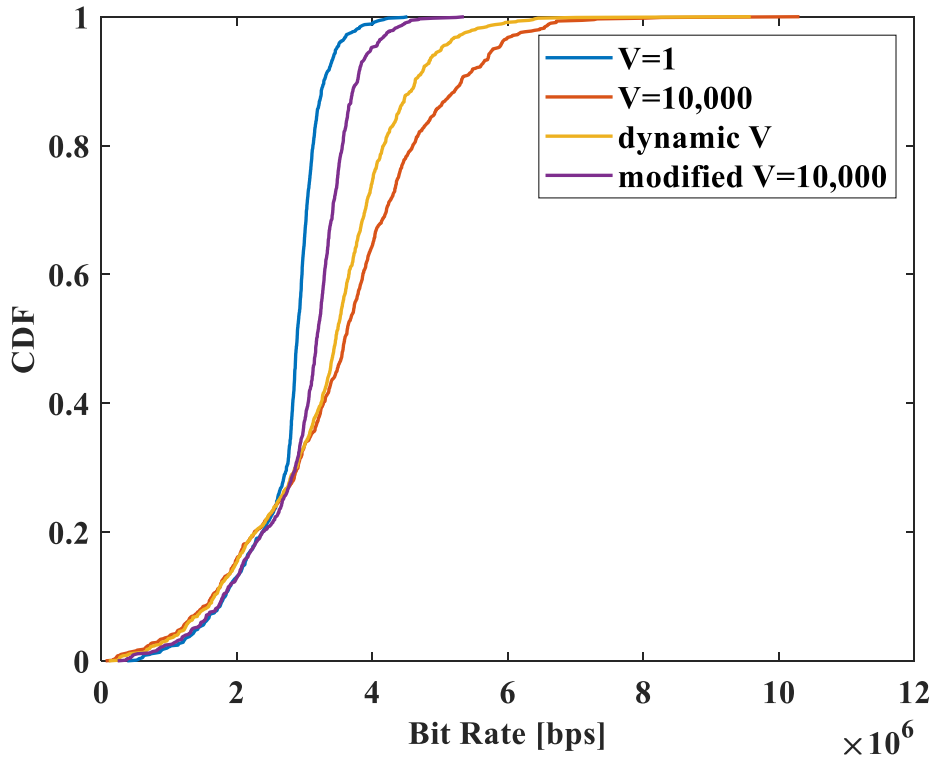


**Figure 7.** CDF of throughput with varying control parameter $V$.

Figure 7 compares the throughput performance of the four scheduling methods. For $V = 1$, the weight of the penalty component reflecting the throughput is very low. Therefore, most UEs with high queue backlogs are scheduled, limiting the maximum throughput performance and resulting in poor throughput performance. For $V = 10{,}000$, the CPU scheduler considers throughput performance, but UEs with $r_k^*(t) > Q_k(t)$ are scheduled, leading to a loss in throughput performance. The dynamic $V$ approach minimizes this throughput loss. It can be observed that dynamic $V$ shows throughput performance that is approximately 90% of $V = 10{,}000$. However, modified $V = 10{,}000$ shows degraded throughput performance in low-traffic scenarios due to not scheduling UEs with $r_k^*(t) > Q_k(t)$, resulting in a significant loss in throughput performance.

### *4.3. SUS Algorithm versus Random Selection*

We compare the SUS algorithm with random selection in terms of queue stabilization performance and throughput performance. To ensure a fair comparison, both methods utilize the dynamic $V$ algorithm and FPA. Figure 8 shows a comparison of queue stabilization performance between the SUS algorithm and random selection, while Figure 9 shows a CDF comparing their throughput performance.

In random selection, when UEs scheduled using Lyapunov optimization exceed the number of antennas at the AP, the UEs are randomly selected to match the number of antennas. UEs scheduled via the SUS algorithm typically have good channel quality and are semiorthogonal to each other. Therefore, when using MR precoding, the overall throughput performance is superior compared to random selection, as depicted in Figure 9. This is advantageous for effectively clearing queue backlogs and maintaining queue stability, particularly in high-traffic scenarios. As shown in Figure 8, the decrease in traffic results in a reduction in the amount of queue backlog. Hence, employing the SUS algorithm for UE scheduling at the AP is essential.

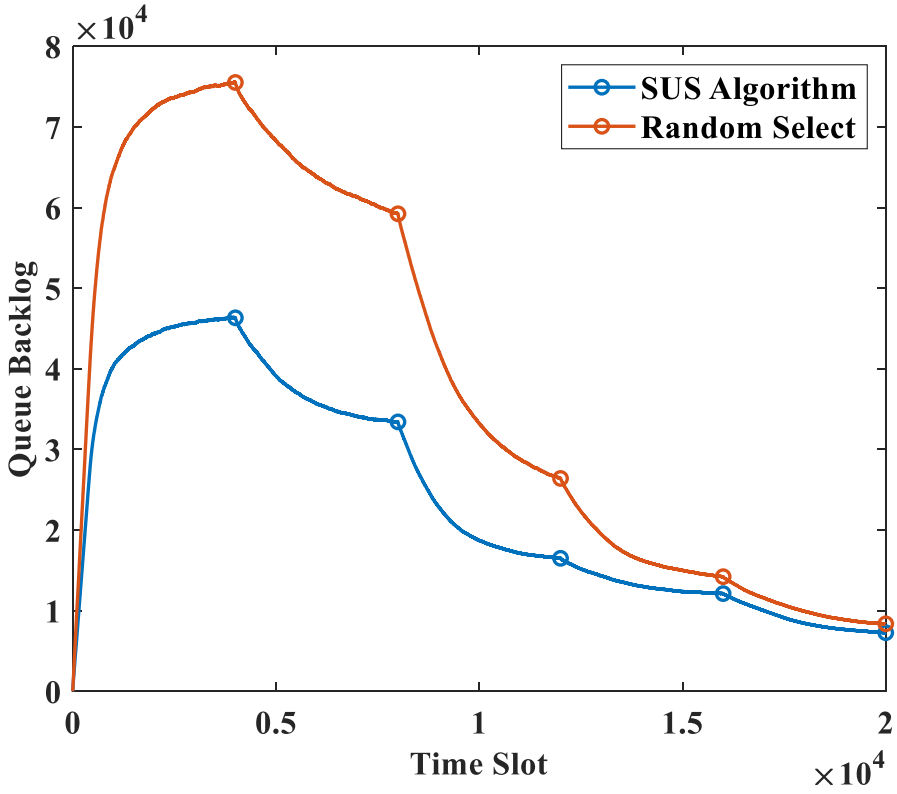


**Figure 8.** Comparison of queue stabilization performance between the SUS algorithm and random selection.

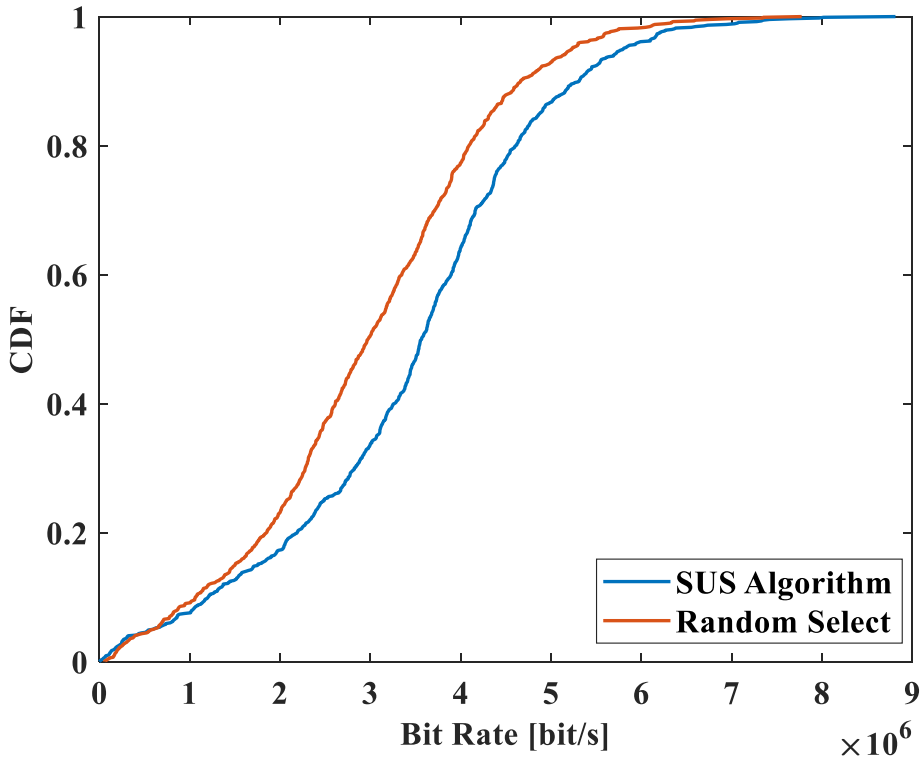


**Figure 9.** CDF comparing throughput performance between the SUS algorithm and random selection.

## 5. Conclusions

This paper proposes a scheduling and power allocation method for dense user-centric scalable CF-MIMO networks. It maintains queue stabilization for each UE while ensuring throughput performance. When Lyapunov optimization in the CPU scheduler is applied, a scheduling method is proposed using the dynamic $V$ approach, exhibiting suitable performance across all traffic intervals. The simulation results show that this scheduling method operates flexibly across all traffic intervals compared to scheduling methods with fixed $V$ values. The issue of UEs exceeding the number of antennas at APs, scheduled using Lyapunov optimization, is addressed by utilizing the SUS algorithm, which reschedules UEs with good channel quality and orthogonality. We propose a power allocation method that combines conventional FPA with the queue-based FPA introduced in this paper. Furthermore, we compare the high-computation complexity MMF and sum SE maximization methods with FPA and FPA + queue-based FPA, which have low complexity. Based on our research, signal processing algorithms combining power allocation and beamforming to reduce latency in dense user-centric scalable CF-MIMO networks will be a future research direction.

**Author Contributions:** Conceptualization, K.-H.S. and J.-W.K.; methodology, K.-H.S. and S.-W.P.; software, K.-H.S. and J.-H.Y.; validation, K.-H.S. and H.-K.S.; formal analysis, K.-H.S.; investigation, K.-H.S. and S.-G.C.; resources, K.-H.S. and H.-D.K.; data curation, K.-H.S.; writing—original draft preparation, K.-H.S.; writing—review and editing, K.-H.S. and H.-K.S.; visualization, K.-H.S.; supervision, Y.-H.Y. and H.-K.S.; project administration, Y.-H.Y. and H.-K.S.; funding acquisition, H.-K.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2020R1A6A1A03038540). This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under the Metaverse Support Program for nurturing the best talents (IITP-2024-RS-2023-00254529), funded by the Korean government (MSIT). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2021-0-01816), supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Zhang, N.; He, P.; Wu, Z.; Chen, P.; Wang, L.; Ye, Z. Latency Analysis and Trial for 5G Ultra Reliable Low Latency Communication. In Proceedings of the 2023 IEEE/CIC International Conference on Communications in China (ICCC Workshops), Dalian, China, 10–12 August 2023; pp. 1–6. [CrossRef]
2. Kim, M.A.; Yoo, S.G.; Kim, H.D.; Shin, K.H.; You, Y.H.; Song, H.K. Group-Connected Impedance Network of RIS-Assisted Rate-Splitting Multiple Access in MU-MIMO Wireless Communication Systems. *Sensors* **2023**, *23*, 3934. [CrossRef]
3. Marzetta, T.L.; Larsson, E.G.; Yang, H.; Ngo, H.Q. *Fundamentals of Massive MIMO*; Cambridge University Press: Cambridge, UK, 2016.
4. Björnson, E.; Hoydis, J.; Sanguinetti, L. Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency. *Found. Trends Signal Process.* **2017**, *11*, 154–655. [CrossRef]
5. Björnson, E.; Sanguinetti, L.; Wymeersch, H.; Hoydis, J.; Marzetta, T.L. Massive MIMO is a Reality—What is Next? Five Promising Research Directions for Antenna Arrays. *arXiv* **2019**, arXiv:1902.07678.
6. Demir, O.T.; Björnson, E.; Sanguinetti, L. Foundations of User-Centric Cell-Free Massive MIMO. *Found. Trends Signal Process.* **2021**, *14*, 162–472. [CrossRef]

7. Ngo, H.Q.; Ashikhmin, A.; Yang, H.; Larsson, E.G.; Marzetta, T.L. Cell-Free Massive MIMO: Uniformly great service for everyone. In Proceedings of the 2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Stockholm, Sweden, 28 June–1 July 2015; pp. 201–205. [CrossRef]
8. Ngo, H.Q.; Ashikhmin, A.; Yang, H.; Larsson, E.G.; Marzetta, T.L. Cell-Free Massive MIMO Versus Small Cells. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1834–1850. [CrossRef]
9. Oh, J.H.; Shin, B.S.; Kim, M.A.; You, Y.H.; Hwang, D.D.; Song, H.K. Efficient User-Serving Scheme in the User-Centric Cell-Free Massive MIMO System. *Sensors* **2022**, *22*, 3794. [CrossRef]
10. Björnson, E.; Sanguinetti, L. Scalable Cell-Free Massive MIMO Systems. *IEEE Trans. Commun.* **2020**, *68*, 4247–4261. [CrossRef]
11. Ammar, H.A.; Adve, R.; Shahbazpanahi, S.; Boudreau, G.; Srinivas, K.V. User-Centric Cell-Free Massive MIMO Networks: A Survey of Opportunities, Challenges and Solutions. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 611–652. [CrossRef]
12. Ammar, H.A.; Adve, R.; Shahbazpanahi, S.; Boudreau, G.; Srinivas, K.V. Downlink Resource Allocation in Multiuser Cell-Free MIMO Networks With User-Centric Clustering. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 1482–1497. [CrossRef]
13. Göttsch, F.; Osawa, N.; Ohseki, T.; Amano, Y.; Kanno, I.; Yamazaki, K.; Caire, G. Fairness Scheduling in Dense User-Centric Cell-Free Massive MIMO Networks. *arXiv* **2022**, arXiv:2211.15294.
14. Neely, M. *Stochastic Network Optimization with Application to Communication and Queueing Systems*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 3. [CrossRef]
15. Wu, Q.; Fan, X.; Wei, W.; Wozniak, M. Dynamic Scheduling Algorithm for Delay-Sensitive Vehicular Safety Applications in Cellular Network. *Inf. Technol. Control* **2020**, *49*, 161–178. [CrossRef]
16. Chen, Z.; Björnson, E.; Larsson, E.G. Dynamic Resource Allocation in Co-Located and Cell-Free Massive MIMO. *IEEE Trans. Green Commun. Netw.* **2020**, *4*, 209–220. [CrossRef]
17. Supittayapornpong, S.; Neely, M.J. Quality of Information Maximization for Wireless Networks via a Fully Separable Quadratic Policy. *IEEE/ACM Trans. Netw.* **2015**, *23*, 574–586. [CrossRef]
18. Bracciale, L.; Loreti, P. Lyapunov Drift-Plus-Penalty Optimization for Queues With Finite Capacity. *IEEE Commun. Lett.* **2020**, *24*, 2555–2558. [CrossRef]
19. Ammar, H.A.; Adve, R.; Shahbazpanahi, S.; Boudreau, G.; Srinivas, K.V. Distributed Resource Allocation Optimization for User-Centric Cell-Free MIMO Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 3099–3115. [CrossRef]
20. Denis, J.; Assaad, M. Improving Cell-Free Massive MIMO Networks Performance: A User Scheduling Approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7360–7374. [CrossRef]
21. Long, R.; Liang, Y.C.; Pei, Y.; Larsson, E.G. Active Reconfigurable Intelligent Surface Aided Wireless Communications. *arXiv* **2021**, arXiv:2103.00709.
22. Chen, Z.; Chang, D.; Tian, Z.; Wang, M.; Zhen, L.; Jia, Y.; Song, J.; Wu, D.O. Joint Power and Pilot Length Allocation for Ultra-Reliable and High-Throughput Transmission. *IEEE Trans. Veh. Technol.* **2024**, 1–13. [CrossRef]
23. Mukherjee, S.; Lee, J. Edge Computing-Enabled Cell-Free Massive MIMO Systems. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2884–2899. [CrossRef]
24. Wen, F.; Gui, G.; Gacanin, H.; Sari, H. Compressive Sampling Framework for 2D-DOA and Polarization Estimation in mmWave Polarized Massive MIMO Systems. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 3071–3083. [CrossRef]
25. Ju, Y.; Cao, Z.; Chen, Y.; Liu, L.; Pei, Q.; Mumtaz, S.; Dong, M.; Guizani, M. NOMA-Assisted Secure Offloading for Vehicular Edge Computing Networks With Asynchronous Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 2627–2640. [CrossRef]
26. Yoo, T.; Goldsmith, A. On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 528–541. [CrossRef]
27. Benmimoune, M.; Driouch, E.; Ajib, W.; Massicotte, D. Joint transmit antenna selection and user scheduling for Massive MIMO systems. In Proceedings of the 2015 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA, USA, 9–12 March 2015; pp. 381–386. [CrossRef]