University of Wollongong
Research Online

University of Wollongong in Dubai - Papers | University of Wollongong in Dubai

January 2020

Pursuit Learning-Based Joint Pilot Allocation and Multi-Base Station Association in a Distributed Massive MIMO Network

Naufan Raharya

Wibowo Hardjawana

Obada Al Khatib
University of Wollongong Dubai

Branka Vucetic

Follow this and additional works at: https://ro.uow.edu.au/dubaipapers

Recommended Citation

Raharya, Naufan; Hardjawana, Wibowo; Al Khatib, Obada; and Vucetic, Branka: Pursuit Learning-Based Joint Pilot Allocation and Multi-Base Station Association in a Distributed Massive MIMO Network 2020, 58898-58911.
https://ro.uow.edu.au/dubaipapers/1145

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 3, 2020, accepted March 12, 2020, date of publication March 24, 2020, date of current version April 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982974

Pursuit Learning-Based Joint Pilot Allocation and Multi-Base Station Association in a Distributed Massive MIMO Network

NAUFAN RAHARYA[1], (Student Member, IEEE), WIBOWO HARDJAWANA[1], (Member, IEEE), OBADA AL-KHATIB[2], (Member, IEEE), AND BRANKA VUCETIC[1], (Life Fellow, IEEE)

[1]School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2008, Australia
[2]Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai 20183, United Arab Emirates

Corresponding author: Naufan Raharya (naufan.raharya@sydney.edu.au)

This work was supported in part by the Australian Research Council Discovery Early Career Research Award under Grant DE150101704, and in part by the Indonesia Endowment Fund for Education (LPDP) through the Postgraduate Scholarship. The work of Branka Vucetic was supported in part by the Australian Research Council Laureate Fellowship under Grant FL160100032.

ABSTRACT Pilot contamination (PC) interference causes inaccurate user equipment (UE) channel estimations and significant signal-to-interference ratio (SINR) degradations. Pilot allocation and multi-base-station (BS) association have been used to combat the PC effect and to maximize the network spectral efficiency. However, current approaches solve the pilot allocation and multi-BS association separately. This leads to a sub-optimal solution. In this paper, we propose a parallel pursuit-learning-based joint pilot allocation and multi-BS association. We first formulate the pilot allocation and multi-BS association problem as a joint optimization function. To solve the optimization function, we use a parallel optimization solver, based on a pursuit learning algorithm, that decomposes the optimization function into multiple subfunctions. Each subfunction collaborates with the other ones to obtain an optimal solution by learning from rewards obtained from probabilistically testing random solution samples. A mathematical proof to guarantee the solution convergence is provided. Simulation results show that our scheme outperforms the existing schemes by an average of 18% in terms of the network spectral efficiency.

INDEX TERMS Multi-BS association, pilot allocation, pursuit learning, pilot contamination, learning automata.

I. INTRODUCTION

The performance of downlink time-division duplex (TDD) wireless multi-cellular networks in massive multiple-input-multiple-output (MIMO) depends on the accuracy of channel state information (CSI) [1], [2]. CSI is acquired from the uplink mutually orthogonal pilot sequences sent by the user equipment (UE) and is used by a base station (BS) to perform downlink beamforming [3]–[6]. The uplink pilot sequences are used at the BS to estimate the CSI of the UE. As the number of pilot sequences is smaller than the number of UEs, multiple UEs might use the same pilot sequences, which causes interference referred to as pilot contamination (PC). The PC causes error in the CSI estimates and these estimation errors result in downlink beamforming inaccuracy, which effectively reduces the downlink signal-to-interference ratio (SINR) for the UEs in a TDD cellular network [7]–[11].

In current works, there are two approaches used to associate a UE with BSs; 1) Cell-free massive MIMO approaches [11], [12] where a UE must be associated with all BSs in the network; and 2) virtual-cell distributed massive MIMO approaches [13]–[27] where a UE associates with a subset of BSs to form a virtual cell. In a cell-free massive MIMO system, there is only a single virtual cell since all UEs must be served by the same BSs. In a virtual-cell system, we can create multiple virtual cells since the UEs are not necessarily served by the same BSs. In real networks, the Channel State Information (CSI) of UEs can only be estimated at a BS if they use different pilot sequences [13]. The number of available pilot sequences is limited by the length of the coherence interval, and the pilot sequences can be reused in different virtual cells. Being a single virtual cell, a cell-free massive MIMO system supports fewer UEs

The associate editor coordinating the review of this manuscript and approving it for publication was Jiayi Zhang.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

compared to a virtual-cell distributed massive MIMO system. This leads to a case where some UEs do not get pilots, referred to as UE outage. The use of virtual cells can avoid this problem; however, in all current virtual-cell distributed massive MIMO system literature, UE outage can still occur because multi-BS association and pilot allocation are considered as separate issues. In a separate process, multi-BS association is done prior to pilot allocation. In the multi-BS association, UEs may share the same BSs. After the pilot allocation, UE outage happens because the number of pilot sequences at the shared BSs is fewer than the number of UEs. The separation of these processes also leads to suboptimal network spectral efficiency.

The optimization solvers for an optimization problem involving multi-BS association and pilot allocation used in [11]–[27] can be divided into two groups. In [11]–[17], the pilot allocation and multi-BS association optimization are solved via a sequential optimization where the solutions are optimized alternately. However, the sequential optimization's complexity grows polynomially with the number of BSs and UEs. The second group in [18]–[27] alleviates the complexity by applying a learning algorithm. The optimization function is solved by using a learning algorithm in parallel such that the optimization procedure is scalable for large number of BSs and UEs. In the learning algorithm, the optimization function is decomposed into multiple sub-functions at the BS or UE level. Each sub-function selects a specific system state that constitutes pilot allocation or BS association configurations for UEs. From the selected system state, the optimization function value can be obtained. Then, this value is used to update the learning function which decides the next system state. The learning process is repeated until the system reaches a stationary state where the learning function satisfies a certain condition.

Based on the system state optimization, existing learning algorithms can be categorized into three types. The first type is Markov decision process (MDP)-based reinforcement learning (RL) that tracks all system states and their state transition probabilities [18]–[20]. The weakness of this approach is the exponential growth of computational complexity due to tracking all possible system states with the number of BSs and UEs. The second type is neural network-based learning that approximates the system states by using a policy function as a learning function that is generated by a neural network [21]–[23]. The policy function represents the best configuration for the optimization function. However, the training of a neural network still requires a large number of system states to be tested and stored in the memory. The third type is learning automata (LA)-based algorithm that uses a probabilistic function as a learning function to determine the next system state. LA eliminates the need to track system states and memory storage [24]–[26]. The commonly used techniques are stochastic learning [24]–[26] and pursuit learning [27]. Stochastic learning uses instantaneous reward to update the probabilistic function, whereas pursuit learning uses the reward history of learning to update the function, which results in significant performance improvements. Unfortunately, the current pursuit learning technique uses a binary reward that cannot be applied in our scenario as the spectral efficiency has a continuous value.

In this paper, we propose a PC mitigation technique based on pursuit learning with a continuous reward value to maximize the network downlink spectral efficiency. We assume that the number of BSs is larger than the number of UEs and the number of UEs is larger than the number of pilot sequences. We first formulate an optimization function of the network spectral efficiency with multi-BS association and pilot allocation as its variables. We then jointly solve the optimization function by applying a pursuit learning algorithm with a continuous reward. First, we treat a communication link between a UE and a BS as an agent whose action is to either probabilistically select a pilot or not. Here, the agent is a decision maker that selects the next system state and an action is one of the possible system states that can be selected by an agent. The environment that evaluates the action selection of the agents is based on the above defined optimization function. The reward, defined as a feedback from the environment, is the calculated network spectral efficiency value for given actions. Here, the value of the reward is continuous. The reward is used to update the average reward history and the action probability. The learning process above is repeated until the action probability converges to unity. We then propose a second pursuit learning algorithm with a heuristic multi-BS association to further improve the speed of learning convergence in terms of the action probability. In the second algorithm, we treat each UE as an agent with an action to select which pilot to be used. Different from the first algorithm, the agent, the UE, is not linked to a BS when choosing an action. In this way, we can reduce the total number of agents and actions from the combination of UE, BS, and pilot sequences to only the combination of UE and pilot sequences. Since we can reduce the number of interacting actions and agents, we can improve the speed of convergence. To decide which BSs a UE is associated with, we use a heuristic approach based on the closest available BSs to UEs. After associating BSs and allocating pilot sequences, we can calculate the network efficiency from the environment and obtain the reward. Then, the learning process proceeds until the action probability converges to unity at each agent. Simulation results show that our proposed scheme yields on average 18% network capacity improvement compared with the best existing schemes. Furthermore, by joining multi-BS association and pilot allocation we can prevent UE outage, defined as a case where some UEs do not get pilots.

The main contributions of this paper are as follows.

1) We propose a joint pilot allocation and multi-BS association optimization. This is unlike [11]–[27], which use separate optimizations. The advantage of the proposed joint optimization over the separate optimizations is a higher spectral efficiency and zero UE outage. In the

separate optimization, the multi-BS association is done prior to the pilot allocation. In the multi-BS association, UEs may share the same BSs. After the pilot allocation, the UE outage happens because the number of pilot sequences at the shared BSs is fewer than the number of UEs.

2) We are the first to develop a parallel pursuit learning approach with a continuous reward, in contrast to the existing pursuit learning approaches that use only a binary reward [27]. Allowing a continuous reward provides a more accurate estimation of the network spectral efficiency of the environment.
3) We incorporate a heuristic solution in the optimization solver, i.e. pursuit learning, to solve the joint optimization function in order to reduce the complexity. By including a heuristic solution in the pursuit learning, we can maintain a high network spectral efficiency while keeping complexity low.
4) We provide a mathematical proof to guarantee that the probabilities of the various actions for the proposed pursuit learning with continuous rewards converge to either 1 or 0. Probability convergence to 1 means that the solution for multi-BS association and pilot allocation is unlikely to change since the probability of selecting the configuration is close to 1. The existing proof in [27] works only for a binary reward and thus cannot be extended to our case.

The remainder of this paper is organized as follows. In Section II, we describe the system model of joint pilot allocation and multi-BS association and formulate an optimization function based on the system model. In Section III, we propose a pursuit learning algorithm as an optimization solver to the optimization function. In Section IV, we incorporate a heuristic solution in the pursuit learning algorithm to reduce the number of iterations required for the algorithm to converge. Section V presents and discusses the numerical results. Section VI concludes the paper.

Notation: Boldface lower and upper case symbols represent vectors and matrices, respectively. The transpose, conjugate-transpose, and conjugate operators are given by $(.)^T$, $(.)^H$, and $(.)^*$ respectively.

II. SYSTEM MODEL

In this section, we first develop the system model for a TDD-based joint pilot and multi-BS association problem. We consider a distributed massive MIMO network where the number of distributed single-antenna BSs (L) is larger than the number of single-antenna UEs (N). In our distributed massive MIMO system, a UE creates its own virtual-cell by associating it with a subset of BSs in the network. Thus, we employ a virtual-cell distributed massive MIMO system. In the system model, the uplink and downlink transmissions are performed in the same spectrum, but in different time slots. We also assume that the number of mutually orthogonal pilot sequences (K) is fewer than the number of UEs.

A. CHANNEL MODEL

We denote $h_{n,l} \in \mathbb{C}^{1\times 1}$, $n = 1, \cdots, N$, $l = 1, \cdots, L$, as the channel gain between BS l and UE n. $h_{n,l}$ is given as

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

$g_{n,l}$ represents short-term fading, which follows an independent and identically distributed (i.i.d) circularly symmetric complex Gaussian distribution, i.e. $\mathcal{CN}(0,1)$. The short-term fading is also assumed to be constant during one coherence interval [28]. $\beta_{n,l}$ is a long-term fading, which changes slowly and can be learned over a long period of time. $\beta_{n,l}$ is modeled as [29]

$$10\log_{10}\beta_{n,l} = -117.8 - 37\log_{10}(d_{n,l}) + \psi, \tag{2}$$

where $d_{n,l}$ represents the distance between UE n and BS l and ψ represents a log-normal random variable, which has a zero mean and σ_{ψ}^2 variance.

B. UPLINK PILOT TRANSMISSION

BSs obtain the CSI through the uplink pilot sequences sent by UEs at the beginning of the coherence interval, representing a time-frequency plane over which the channel is static [28]. We denote $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_K] \in \mathbb{C}^{K\times K}$ as the set of K orthogonal pilot sequences, e.g. $\mathbf{\Phi}^{\boldsymbol{H}}\mathbf{\Phi} = \mathbf{I_K}$, available at each BS, and each pilot has the length of K symbols $\boldsymbol{\phi}_k = \left[\phi_{k1}, \cdots, \phi_{kK}\right]^T$, $k = 1, \cdots, K$. We then denote $\boldsymbol{\varphi}_n \in \mathbb{C}^{K\times 1}$ as the pilot for UE n, chosen from $\mathbf{\Phi}$, i.e. $\boldsymbol{\varphi}_n \in \mathbf{\Phi}$, by UE n. Then, the association variable for pilot k of BS l to UE n, $v_{n,l}^k$, is given as follows

$$v_{n,l}^k = \begin{cases} 1, & \text{if BS } l \text{ allocates pilot } k \text{ to UE } n, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

BS l receives the uplink pilot k as, $\mathbf{y}_{l,k}^p \in \mathbb{C}^{K\times 1}$. It is given by

$$\mathbf{y}_{l,k}^p = \sqrt{\rho_p}\sum_{\check{l}=1}^{L}\sum_{n=1}^{N} h_{n,l} v_{n,\check{l}}^k \boldsymbol{\varphi}_n + \mathbf{z_l}, \tag{4}$$

where $\mathbf{z_l} \in \mathbb{C}^{K\times 1}$ i.i.d $\mathcal{CN}(0, 1)$ is the additive white Gaussian noise at BS l and $\sqrt{\rho_p}$ is the uplink pilot power from UE n. BS l first multiplies $\mathbf{y}_{l,k}^p$ with the pilot sequence of UE n, i.e. $\boldsymbol{\varphi}_n$, to obtain $\tilde{y}_{n,l}^k$, which denotes the CSI of UE n when using pilot k,

$$\begin{aligned}\tilde{y}_{n,l}^k &= \boldsymbol{\varphi}_n^H \mathbf{y}_{l,k}^p \\ &= \sqrt{\rho_p} h_{n,l} v_{n,l}^k + \sqrt{\rho_p}\sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N}\sum_{\substack{\acute{l}=1 \\ \acute{l}\neq l}}^{L} h_{\acute{n},l} v_{\acute{n},\acute{l}}^k \boldsymbol{\varphi}_n^H \boldsymbol{\varphi}_{\acute{n}} + \tilde{z}_l. \end{aligned} \tag{5}$$

The first-term of (5) is the channel gain of UE n to BS l using pilot k. On the other hand, the second-term of the equation represents the channel gain from other UEs using the same pilot k assigned by other BSs. This term is referred to as the *pilot contamination* (PC). The last term is the product of the BS receiver noise and the pilot sequence, i.e. $\tilde{z}_l \triangleq \boldsymbol{\varphi}_n^H \mathbf{z}_l$.

The minimum mean square error (MMSE) of UE n at BS l when using pilot k for a given $\tilde{y}_{n,l}^k$ is expressed by [30]

$$\hat{h}_{n,l}^k = \kappa_{n,l}^k \tilde{y}_{n,l}^k, \tag{6}$$

where $\kappa_{n,l}^k$ is given as follows

$$\begin{aligned}\kappa_{n,l}^k &= \frac{\mathrm{E}\{\tilde{y}_{n,l}^{kH} h_{n,l}\}}{\mathrm{E}\{|\tilde{y}_{n,l}^{kH}|^2\}} \tilde{y}_{n,l}^k \\ &= \frac{\sqrt{\rho_p}\beta_{n,l} v_{n,l}^k}{\rho_p \sum_{n=1}^{N}\sum_{\check{l}=1}^{L} \beta_{n,\check{l}} v_{n,\check{l}}^k \boldsymbol{\varphi}_n^H \boldsymbol{\varphi}_{\acute{n}} + \sigma_{z_l}^2} \tilde{y}_{n,l}^k \end{aligned} \tag{7}$$

C. DOWNLINK DATA TRANSMISSION

In the downlink data transmission, we adopt the linear maximum ratio transmission (MRT) due to its simplicity and robustness compared to other precoding schemes in large-scale antenna systems [31]. We denote by $\eta_{n,l}^k \in \mathbb{C}^{1\times 1}$ the transmitted downlink signal to UE n that uses pilot k of BS l. $\eta_{n,l}^k$ is given by

$$\eta_{n,l}^k = \sqrt{\alpha_{n,l}^k}\hat{h}_{n,l}^{kH} v_{n,l}^k q_n, \tag{8}$$

where q_n is the symbol sent to UE n from BS l, which is independent from noise and channel gain, and $\alpha_{n,l}^k$ is the allocated downlink transmit power at BS l for UE n when using pilot k, given as

$$\alpha_{n,l}^k = \frac{p_l^d v_{n,l}^k}{\sum_{n=1}^{N}\sum_{k=1}^{K} \mathrm{E}\{|\hat{h}_{n,l}^k|^2\}}. \tag{9}$$

The total transmitted signal at l in (8) is allocated proportionally to UEs according to the channel strength and must not exceed the total transmitted downlink power at BS l, p_l^d, given as

$$\mathrm{E}\{|\sum_{n=1}^{N}\sum_{k=1}^{K} \eta_{n,l}^k|^2\} \le p_l^d. \tag{10}$$

Finally, the received signal for each UE n is given by

$$y_n^d = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{\acute{n}=1}^{N} \sqrt{\alpha_{n,l}^k}\hat{h}_{\acute{n},l}^{kH} h_{n,l} q_{\acute{n}} v_{\acute{n},l}^k + w_n. \tag{11}$$

We assume that UE n is only aware of the statistics of the estimated channel, i.e. $\mathrm{E}\{|\hat{h}_{n,l}^k|^2\} = \sqrt{\rho_p}\beta_{n,l}\kappa_{n,l}^k$, [11], [28], [32]. Thus, (11) can be written as

$$\begin{aligned} y_n^d = &\underbrace{\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\mathrm{E}\{\hat{h}_{n,l}^{kH} h_{n,l} v_{n,l}^k\} q_n}_{S_A} \\ &+ \underbrace{\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\big(\hat{h}_{n,l}^{kH} h_{n,l} v_{n,l}^k - \mathrm{E}\{\hat{h}_{n,l}^{kH} h_{n,l} v_{n,l}^k\}\big) q_n}_{S_C} \\ &+ \underbrace{\sum_{k=1}^{K}\sum_{\substack{\acute{n}=1\\ \acute{n}\neq n}}^{N}\sum_{l=1}^{L}\sqrt{\alpha_{\acute{n},l}^k}\hat{h}_{\acute{n},l}^{kH} h_{n,l} v_{\acute{n},l}^k q_{\acute{n}} + w_n}_{S_B} \end{aligned} \tag{12}$$

The received signal in (12) consists of the desired signal, the interference and noise [11], [33]. S_A represents the desired signal from BSs that serves UE n in pilot k and S_B represents a multi UE interference. The second term, S_C is treated as the self-interference and resulted from the lack of channel realization knowledge at the UE's receiver.

Based on (12), SINR of UE n is given as follows [11]

$$\gamma_n = \frac{\mathrm{E}\{|S_A|^2\}}{\mathrm{E}\{|S_B|^2\} + \mathrm{E}\{|S_C|^2\} + \sigma_{w_n}^2}, \tag{13}$$

where $\sigma_{w_n}^2$ is the variance of UE's noise w_n. $\mathrm{E}\{|S_A|^2\}$, $\mathrm{E}\{|S_B|^2\}$, and $\mathrm{E}\{|S_C|^2\}$ are given as follows

$$\mathrm{E}\{|S_A|^2\} = \rho_p\Big(\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\kappa_{n,l}^k\beta_{n,l}v_{n,l}^k\Big)^2, \tag{14}$$

$$\begin{aligned}\mathrm{E}\{|S_B|^2\} = &\rho_p\sum_{\substack{\acute{n}=1\\ \acute{n}\neq n}}^{N}\Big(\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{\acute{n},l}^k}\kappa_{\acute{n},l}^k\beta_{n,l}v_{\acute{n},l}^k\Big)^2 \\ &\times |\boldsymbol{\varphi}_{\acute{n}}^H\boldsymbol{\varphi}_n|^2 + \sqrt{\rho_p}\sum_{k=1}^{K}\sum_{\substack{\acute{n}=1\\ \acute{n}\neq n}}^{N}\sum_{l=1}^{L}\alpha_{\acute{n},l}^k\kappa_{\acute{n},l}^k\beta_{\acute{n},l} \\ &\times \beta_{n,l}v_{\acute{n},l}^k, \end{aligned} \tag{15}$$

$$\mathrm{E}\{|S_C|^2\} = \sqrt{\rho_p}\sum_{k=1}^{K}\sum_{l=1}^{L}\alpha_{n,l}^k\kappa_{n,l}^k\beta_{n,l}^2 v_{n,l}^k. \tag{16}$$

The first term of (15) is from PC interference and the second term is from the other UE's channel interference. (14)-(16) are derived based on the product rules of Gaussian multiple random variables in [34] and [11], given as follows

$$\mathrm{E}\{|g_{n,l}|^{2f}\} = f!(\mathrm{E}\{|g_{n,l}|^2\})^f, \quad f = 1, 2, 3, \cdots \tag{17}$$

$$\mathrm{E}\{(g_{n,l}g_{\acute{n},l})\} = 0, \tag{18}$$

$$\mathrm{E}\{(g_{n,l}g_{\acute{n},l})^2\} = 1. \tag{19}$$

The complete proof of (14)-(16) is given in Appendix A-C.

D. PROBLEM FORMULATION

Based on (13), the SINR depends on the variable $v_{n,l}^k$, which represents the pilot allocation and BSs association with a UE. Subsequently, we can formulate an optimization function that aims to maximize the network spectral efficiency as follows

$$\underset{v_{n,l}^k}{\text{maximize}} U = \sum_{n=1}^{N}\log_2\Big(1+\gamma_n\Big) \tag{20}$$

$$\text{subject to } v_{n,l}^k = \{0, 1\}, \tag{21}$$

$$\sum_{k=1}^{K}\tilde{v}_n^k \le 1, \quad \forall n, \tag{22}$$

$$\sum_{n=1}^{N} v_{n,l}^k \le 1, \quad \forall l, \ \forall k, \tag{23}$$

$$\sum_{k=1}^{K}\sum_{n=1}^{N} v_{n,l}^k \le K, \quad \forall l, \tag{24}$$

$$\sum_{k,l=1}^{K,L} v_{n,l}^{k} \geq 1, \quad \forall n, \tag{25}$$

where $\tilde{v}_n^k$ is given by

$$\tilde{v}_n^k = \begin{cases} 1, & \text{if } \sum_{l=1}^{L} v_{n,l}^{k} \neq 0, \ \forall n, \ \forall k, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

(26) indicates whether pilot k is used by UE n. If pilot k is used by UE n, the value of $\tilde{v}_n^k$ will be 1 and it is obtained by doing a summation of $v_{n,l}^k$ over L BSs, i.e. $\sum_{l=1}^{L} v_{n,l}^k$. Based on (26), (22) restricts UE n to use a maximum of one pilot sequence only when it is associated with any BS. (23) means that pilot k at BS l can only be used by a maximum of one UE and (24) denotes a constraint that only allows a BS to serve a maximum of K UEs. Finally, (25) denotes that all UEs must at least be served by one BS and allocated a pilot.

Note that we could use our virtual-cell distributed massive MIMO system to model a cell-free massive MIMO system, where each UE must be served or connected to all available BSs. This is done by including the following constraint in the optimization problem given by (20),

$$\sum_{k,l=1}^{K,L} v_{n,l}^{k} = L, \quad \forall n, \tag{27}$$

where, $v_{n,l}^k$, K and L denote the association variable for UE n to pilot k at BS l, the number of pilot sequences and distributed BSs, respectively

III. PURSUIT LEARNING WITH CONTINUOUS REWARD

In this section, we formulate a pursuit learning algorithm with a continuous reward to solve (20) and provide a convergence proof for the algorithm.

A. PURSUIT LEARNING ELEMENTS AND ALGORITHM

In this subsection, we first define the pursuit learning elements that construct the pursuit learning algorithm. Then, we show how to solve (20) by using the defined pursuit learning elements.

1) AGENT

An agent is a decision maker that selects the next action through a feedback from the system. We treat the communication link n, l between UE n and BS l as agent $n, l, n \in \{1, \cdots, N\}, l \in \{1, \cdots, L\}$.

2) ACTION

An action is one of the possible system states that can be chosen by an agent. We denote $\mathbf{a} = \{a_{1,1}(t), \cdots, a_{N,1}(t), \cdots, a_{N,L}(t)\}$, where $a_{n,l}(t) = k, k = 0, 1, \cdots, K$, is an action by agent n, l at step t to either not select pilot, $k = 0$, or select a pilot, $k = 1, \ldots, K$. $a_{n,l}(t)$ translates into optimization variable $v_{n,l}^k$ given as follows

$$v_{n,l}^{k} = \begin{cases} 1, & \text{if } k = a_{n,l}(t), a_{n,l}(t) \neq 0, \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

3) PROBABILITY VECTOR

A probability vector consists of the probabilities of all actions to be selected by agent n, l at step t, denoted by $\mathbf{p}_{n,l}(t) = \{p_{n,l}^0(t), p_{n,l}^1(t), \cdots, p_{n,l}^K(t)\}$. The sum of all probability elements in $\mathbf{p}_{n,l}(t)$ must be equal to one,

$$\sum_{k=0}^{K} p_{n,l}^{k}(t) = 1. \tag{29}$$

4) ENVIRONMENT

The environment is a place where $v_{n,l}^k$ configurations in (20) are evaluated. (21)-(25) are the environment for agents in the optimization problem.

5) REWARD

The reward is the feedback from the environment as a response to $a_{n,l}(t), \forall n, l$. The reward at step t, $R(t)$ is given by

$$R(t) = U(t) \times J(t) \tag{30}$$

where $U(t)$ is the value of U in (20) based on the value of $v_{n,l}^k$, according to actions $a_{n,l}(t)$ above. $J(t)$ is an indicator function if the constraints in (21)-(25) are satisfied for given $v_{n,l}^k$,

$$J(t) = \begin{cases} 1, & \text{if (21)-(25) are satisfied} \\ -A, & \text{otherwise} \end{cases}, \tag{31}$$

where A is a positive number, i.e. $A > 1$. Note that $R(t)$ has a continuous reward as it is a function of $U(t)$. This is in contrast to the binary reward in [27].

6) AVERAGE REWARD HISTORY

The average reward history is defined as the mean of the rewards of selecting actions until step t. The average reward history is a unique feature of a pursuit learning algorithm. This feature is not used in the stochastic learning algorithm in [24]–[26]. The average reward denotes the average reward of action k by agent n, l at step t and it can be represented as $r_{n,l}^k(t) \in \mathbf{r}_{n,l}(t) = \{r_{n,l}^0(t), r_{n,l}^1(t), \cdots, r_{n,l}^K(t)\}$,

$$r_{n,l}^{k}(t) = \begin{cases} r_{n,l}^{k}(t-1) + \dfrac{R(t) - r_{n,l}^{k}(t-1)}{c_{n,l}^{k}(t)}, & \text{if } k = a_{n,l}(t), \\ r_{n,l}^{k}(t-1), & \text{if } k \neq a_{n,l}(t) \end{cases}, \tag{32}$$

where $c_{n,l}^k(t) \in \mathbf{c}_{n,l} = \{c_{n,l}^0, c_{n,l}^1, \cdots, c_{n,l}^K\}$, is the occurrence or the number of times action k is selected by agent n, l at step t. $c_{n,l}^k(t)$ is given as follows

$$c_{n,l}^{k}(t) = \begin{cases} c_{n,l}^{k}(t-1) + 1, & \text{if } k = a_{n,l}(t), \\ c_{n,l}^{k}(t-1), & \text{if } k \neq a_{n,l}(t) \end{cases} \tag{33}$$

The action that gives the maximum reward history in $\mathbf{r}_{n,l}(t)$ is denoted by $m_{n,l}$ and is given as

$$m_{n,l} = \underset{k \in \{0,1,\cdots,K\}}{\operatorname{argmax}} \{\mathbf{r}_{n,l}(t)\}. \tag{34}$$

(34) is then used to update the probability vector $\mathbf{p}_{n,l}(t)$,

$$\mathbf{p}_{n,l}(t) = (1-\theta)\mathbf{p}_{n,l}(t-1) + \theta \mathbf{e}_{m_{n,l}}(t), \tag{35}$$

where θ is a learning step and its value has a range $0 < \theta < 1$. $\mathbf{e}_{m_{n,l}}(t)$ is a binary vector whose length is the same as the length of the probability vector $\mathbf{p}_{n,l}$ and $m_{n,l}$-th entry is 1 (the other entries are 0).

In the first line of Alg. 1, we initialize the values of $R(t)$, $p^k_{n,l}(t)$, $r^k_{n,l}(t)$, and $c^k_{n,l}(t)$. Inside the loop, at step t, agent n, l first selects its action based on probability vector $\mathbf{p}_{n,l}(t)$. The selected action by agent n, l at step t is represented by $a_{n,l}(t) = k$. $v^k_{n,l}$ is then obtained from (28) and used to calculate $U(t)$ (e.g., the value of U at step t). The reward and subsequently the average reward history are obtained from (30) and (32). The action probability of agent n, l is updated by using (35). The learning process is repeated until there is an action probability of each agent that is larger than 0.99.

Algorithm 1 Pursuit Learning Algorithm

1: *Initialization*: set $t = 0$, $R(0) = 0$, $p^k_{n,l}(0) = \frac{1}{K+1}$, $r^k_{n,l}(0) = 0$, $c^k_{n,l}(0) = 0$.
2: **Loop for** $t = 1, 2, 3, \cdots$,
3: At step t, agent n, l selects their action, based on the probability vector as in $\mathbf{p}_{n,l}(t)$, i.e. $a_{n,l}(t) = k$, $k \in \{0, 1 \cdots, K\}$.
4: The selected action in previous step sets $v^k_{n,l}$ into one or zero as given in (28) and then the value in (20) is obtained.
5: The reward for all agents are calculated in (30) by substituting $U(t)$ with the value in (20) and calculating (31).
6: Then, each agent updates their own average reward history by using (32) and (33).
7: Agent n, l finds the action that yields the maximum value in $\mathbf{r}_{n,l}(t)$ by using (34).
8: Then, the probability is updated by using (35).
9: **Stop** if there is a probability $p^k_{n,l}(t)$ in $\mathbf{p}_{n,l}(t)$, $\forall n, l$ that is larger than 0.99.

B. PROBABILITIES CONVERGENCE PROOF

The pursuit learning algorithm as constructed in (35) shows that if there is an action $m_{n,l}$ for agent n, l, for which its average reward history, $r^{m_{n,l}}_{n,l}(t)$, stays maximum after step t, $p^{m_{n,l}}_{n,l}(t)$ converges to 1.

Theorem 1: After step t, $t > t_0$, there is an action $m_{n,l}$ that yields the maximum average reward history $r^{m_{n,l}}_{n,l}(t)$ in $\mathbf{r}_{n,l}(t)$, such that

$$r^{m_{n,l}}_{n,l}(t) > r^k_{n,l}(t), \ k \neq m_{n,l}, \tag{36}$$

$$r^{m_{n,l}}_{n,l}(t) \in \mathbb{R}, \tag{37}$$

then,

$$\lim_{t\to\infty} p^{m_{n,l}}_{n,l}(t) \to 1. \tag{38}$$

Proof: Action $m_{n,l}$ of agent n, l has the largest reward history as given in (34). Thus, based on (35), the probability of agent n, l selecting action $m_{n,l}$ at step $t+1$ and step t can be expressed as

$$p^{m_{n,l}}_{n,l}(t+1) = (1-\theta)p^{m_{n,l}}_{n,l}(t) + \theta. \tag{39}$$

We then define the difference of value between the probability of choosing action $m_{n,l}$ by agent n, l at step $t+1$, $p^{m_{n,l}}_{n,l}(t+1)$, and the probability of choosing action $m_{n,l}$ by agent n, l at step t, $p^{m_{n,l}}_{n,l}(t)$ as

$$\Delta p^{m_{n,l}}_{n,l}(t+1) = p^{m_{n,l}}_{n,l}(t+1) - p^{m_{n,l}}_{n,l}(t). \tag{40}$$

By using (39) and (40), we can obtain

$$\begin{aligned}\Delta p^{m_{n,l}}_{n,l}(t+1) &= \theta(1 - p^{m_{n,l}}_{n,l}(t)) + p^{m_{n,l}}_{n,l}(t) - p^{m_{n,l}}_{n,l}(t) \\ &= [1 - p^{m_{n,l}}_{n,l}(t)]\theta. \end{aligned} \tag{41}$$

As $0 \leq p^{m_{n,l}}_{n,l}(t) \leq 1, \ t = 1, \cdots, \infty$, therefore:

$$\Delta p^{m_{n,l}}_{n,l}(t+1) \geq 0. \tag{42}$$

By applying (42) to (40), we then get

$$p^{m_{n,l}}_{n,l}(t+1) \geq p^{m_{n,l}}_{n,l}(t). \tag{43}$$

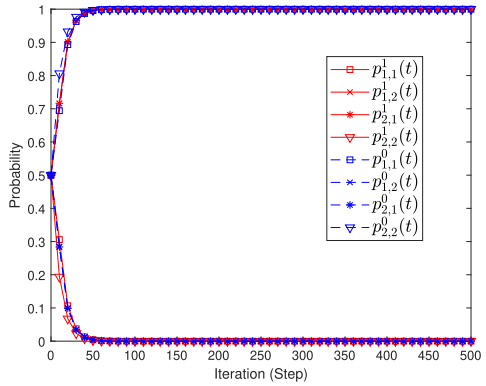
(43) guarantees condition in (38). This completes the proof.

C. CONVERGENCE BEHAVIOR

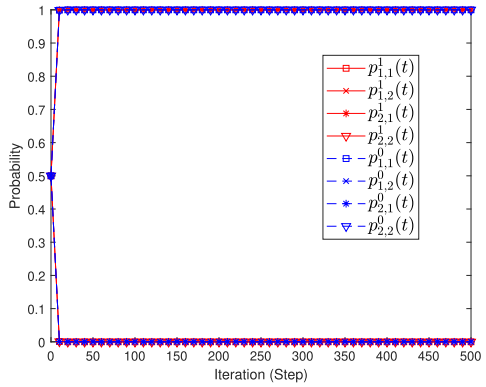
In this section, we show the probability convergence behavior of actions of an agent by using a numerical simulation. For this simulation, we have 2 UEs, 2 BSs, and 1 pilot sequence. We also set the value of A in (31) to be 10. The probability convergence behavior for the example above is shown in Fig. 1 for different values of θ. In the figure, the subscript of p denotes an index for agent n, l and the superscript denotes an index for action k of agent n, l. For instance, $p^1_{1,2}(t)$ denotes the probability of agent 1, 2 selecting action 1. Initially, all action probabilities of an agent are equal to 1/2, i.e. $p^1_{1,1}(0) = 1/2$ and $p^0_{1,1}(0) = 1/2$. As the iteration step increases, one action probability of each agent converges to 1. The probability convergence results in Fig. 1 support the proof of Theorem 1 for different values of θ. Note that the trade-off between the speed of convergence and the optimization value for different θ will be discussed further in Section V.

IV. HEURISTIC SOLUTION

In this section, we aim to reduce the number of iterations required for the action probability to converge to 1. To do this we introduce a heuristic method to associate UEs with BSs in the proposed pursuit learning algorithm. In computing the association between UEs and BSs, the heuristic method assumes each UE has chosen a pilot sequence and selects BSs to associate with based on long-term channel gain $\beta_{n,l}$ and constraints (21)-(25) shown in (20). The selected pilot sequences and BSs are used to obtain the optimization variable $v^k_{n,l}$ which is then used to compute (20). The reward is



(a) $\theta = 0.1$



(b) $\theta = 0.5$

FIGURE 1. Evolution of action selection probabilities for $N = 2, L = 2, K = 1$.

obtained from this step and it is used to update the average reward history and the probability vector. The next action of the UE to select a pilot sequence is based on the updated probability vector. We explain now in detail how we modify the pursuit learning algorithm in Section III to do the above.

First, we reduce the number of agents and actions. We treat UE $n \in \{1, \cdots, N\}$ as agent n. Note that the agent is changed from the link between UE n and BS l (agent n, l) in Alg. 1 to UE n (agent n). We can then drop index l in $a_{n,l}(t)$ and denote $a_n(t) = k$ as an action by agent n to select pilot $k \in \{1, \cdots, K\}$ at step t. By doing the above, the total number of agents and actions is reduced from $NL(K+1)$ to NK. Reducing the number of interacting actions and agents then improves the speed of convergence of the action probability to unity.

Note that the mentioned heuristic method for UEs to select BSs to associate with, requires information on which pilot UE uses. Therefore, in the beginning, we need to set initial pilot selections for UE. To do this, agent n selects pilot k according to the probability vector. The pilot selection sets $a_n = k$ and it means that agent or UE n will use pilot k.

Once the pilot assignment for each UE is obtained, we perform the proposed heuristic method to obtain the association configurations of UEs to BSs. To do this, we start with the first BS, i.e. $l = 1$. At the BS, we begin with agent $\hat{n}_1$ that denotes the agent with the lowest average long-term

Algorithm 2 Heuristic-Pursuit Learning Algorithm

Set $t = 0$, $R(0) = 0$, $p^k_n(0) = 1/K$, $r^k_n(0) = 0$, $c^k_n(0) = 0$, $v^k_{n,l} = 0$.
Loop for $t = 1, 2, 3, \cdots$,
At step t, agent n selects an action according to probability vector $\mathbf{p}_n(t)$, i.e. $a_n(t) = k$.
for $l = 1$ to L **do**
  for $n^* = 1$ to N **do**
    if $\sum_{k=1}^{K}\sum_{n=1}^{N} v^k_{n,l_{\hat{n}_{n^*}}} < K$ and $\sum_{n=1}^{N} v^{a_{\hat{n}_{n^*}}}_{n,l_{\hat{n}_{n^*}}} = 0$ **then**
      $v^{a_{\hat{n}_{n^*}}}_{\hat{n}_{n^*},l_{\hat{n}_{n^*}}} = 1$.
    else if $l = 1$ **then**
      agent n selects the next closest BS, e.g. $l_{\hat{n}_{n^*}} + 1$, that satisfies $\sum_{k=1}^{K}\sum_{n=1}^{N} v^k_{n,l_{\hat{n}_{n^*}}} < K$ and $\sum_{n=1}^{N} v^{a_{\hat{n}_{n^*}}}_{n,l_{\hat{n}_{n^*}}} = 0$. It thus modifies $v^{a_{\hat{n}_{n^*}}}_{\hat{n}_{n^*},l^*_{\hat{n}_{n^*}}} = 1$ with $l^*_{\hat{n}_{n^*}}$ is the selected BS.
    else
      continue
    end if
  end for
end for
$v^k_{n,l}$ is used to calculate (20) and (30) is obtained.
Then, each agent updates their own average reward history by using (32) and (33).
Agent n finds an action that yields a maximum value in $\mathbf{r}_n(t)$ by using (34).
Then, the probability is updated by using (35).
Stop if there is a probability $p^k_n(t), k \in \{1, \cdots, k\}, \forall n$, larger than 0.99.

channel gain, i.e. $\frac{\sum_{l=1}^{L}\beta_{\hat{n}_1,l}}{L} < \frac{\sum_{l=1}^{L}\beta_{\hat{n}_2,l}}{L} < \cdots < \frac{\sum_{l=1}^{L}\beta_{\hat{n}_N,l}}{L}$, $\hat{n}_n = 1, \cdots, N$. Agent $\hat{n}_1$ then verifies the first BS in $\mathbf{l}_{\hat{n}_1}$, where $\mathbf{l}_{\hat{n}_n} = \{1_{\hat{n}_n}, \cdots, L_{\hat{n}_n}\}$ and $l_{\hat{n}_n}$ satisfies $\beta_{\hat{n}_n,1_{\hat{n}_n}} > \beta_{\hat{n}_n,2_{\hat{n}_n}} > \cdots > \beta_{\hat{n}_n,L_{\hat{n}_n}}$, whether it can satisfy $\sum_{k=1}^{K}\sum_{n=1}^{N} v^k_{n,l_{\hat{n}_1}} < K$, which relates to (23), and $\sum_{n=1}^{N} v^{a_{\hat{n}_1}}_{n,l_{\hat{n}_1}} = 0$, which means whether the particular selected pilot k at BS $l_{\hat{n}_1}$ has been associated with other agents or not. Then, the selection sets $v^k_{n,l} = 1$ for $n = \hat{n}_1$, $l = l_{\hat{n}_1}, k = a_{\hat{n}_1}$. When agent $\hat{n}_1$ cannot satisfy the previous constraint and $l_{\hat{n}_n}$ is its first BS, it selects the next BS, e.g. $l_{\hat{n}_1} + 1$, that still serves less than K UEs and is not associated with other UEs for a selected pilot k. It also sets $v^k_{n,l} = 1$ for $n = \hat{n}_1, l = l^*_{\hat{n}_1}, k = a_{\hat{n}_1}$, where $l^*_{\hat{n}_1}$ is the selected BS. Otherwise, $v^{a_{\hat{n}_1}}_{\hat{n}_1,l_{\hat{n}_1}} = 0$. The above process is repeated by agent $\hat{n}_2$ to $\hat{n}_N$ and at $l = 2$ to $l = L$. The $v^k_{n,l}$ configurations are now completed.

Once the heuristic method is completed we then calculate the rewards for the UEs to select the pilots. This is done by calculating the value of $R(t)$ in (30) with its $U(t)$ in (20) and $J(t)$ in (31) respectively. $R(t)$ is then used to update (32)-(35) by replacing $r^k_{n,l}(t)$, $c^k_{n,l}(t)$, $m_{n,l}$, and $\mathbf{p}_{n,l}(t)$ with

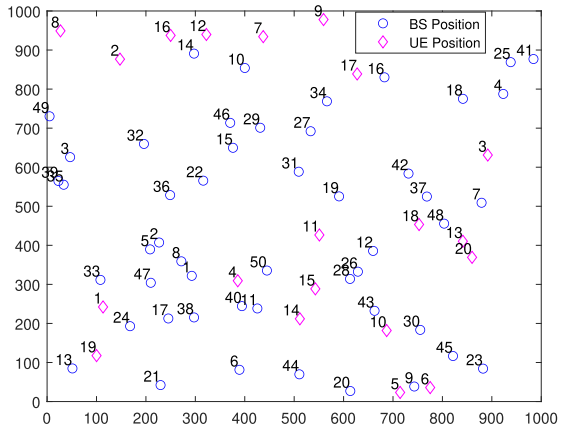


FIGURE 2. Deployment of BSs and UEs in the simulation with $L = 50$ and $N = 20$.

$r_n^k(t) \in \mathbf{r}_n(t) = \{r_n^1(t), \cdots, r_n^K(t)\}$, $c_n^k(t) \in \mathbf{c}_n(t) = \{c_n^1(t), \cdots, c_n^K(t)\}$, m_n, and $\mathbf{p}_n(t) = \{p_n^1, \cdots, p_n^k\}$, respectively. After updating the above elements, the pursuit learning is repeated until there is $p_n^k(t), \forall n$, larger than 0.99. The heuristic-pursuit learning scheme is given in Alg. 2.

V. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed scheme, and compare it to other known schemes. We consider a square area of length 1000 m where BSs and UEs are uniformly distributed within the area. Fig. 2 illustrates the deployment of BSs and UEs in the network in our simulation. We simulate 300 independent realizations by varying the positions of BSs and UEs. We also make the square area wrap around the edges to avoid the boundary effect.

Table 1 summarizes the setup parameters. In addition, we compare our result with other schemes whose legends are defined as follows

1) Alg. 1: This scheme refers to the use of Alg. 1 to solve (20) as described in Section III.
2) Alg. 2: This scheme refers to the use of Alg. 2 to solve (20).
3) Sch in [15]: Scheme (Sch) in [15] does a separate optimization between the pilot allocation and the multi-BS association. The multi-BS association is done prior to the pilot allocation by using the following equation:

$$\sum_{l=1}^{L_{0,n}} \frac{\hat{\beta}_{n,l}}{\sum_{\acute{l}=1}^{L} \beta_{\acute{l},n}} \leq \delta\% \tag{44}$$

where $L_{0,n} \leq L$ denotes the number associated BSs with UE n and $\{\hat{\beta}_{n,1}, \cdots, \hat{\beta}_{n,L}\}$ is the sorted (in a descending order) set of $\{\beta_{n,1}, \cdots, \beta_{n,L}\}$. The value of δ is chosen such that the network spectral efficiency is maximized and the UE outage is minimized for given pilots, BSs, and UEs. Note that, the UE outage happens because the number of pilot sequences at the associated BSs is smaller than the number of UEs. In addition, if UE n cannot satisfy $\delta\%$, it is associated with a BS with the strongest $\beta_{n,l}$. After the multi-BSs association, this scheme allocates pilots by using a sequential optimization technique [15].
4) Joint SL (Stochastic Learning): This scheme jointly solves the pilot allocation and the multi-BS association by using stochastic learning. The stochastic learning solves the optimization problem without using the average reward history.
5) Sch in [24]: This scheme separates the pilot allocation and the multi-BS association. The multi-BS association is done by following (44) and the pilot allocation is done by using stochastic learning.

TABLE 1. Simulation parameters.

Parameters	Value
Area length (square)	1000 m
System bandwidth	10 MHz
Carrier frequency	900 MHz
Number of pilots (K)	4
Number of UEs (N)	20
Maximum power of AP (P_{max})	13 dBm
Pilot transmission power (ρ_p)	0 dBm
White noise power density	-174 dBm/Hz
A	10

TABLE 2. Complexity comparison.

Algorithm	Complexity
Alg. 1	$\mathcal{O}\Big((K+1)i_1\Big)$
Alg. 2	$\mathcal{O}\Big((K+NL)i_2\Big)$
Sch in [15]	$\mathcal{O}\Big(NL + i_3(N^2LK + N^2)\Big)$
Joint SL	$\mathcal{O}\Big((K+1)i_4\Big)$
Sch in [24]	$\mathcal{O}\Big(NL + i_5(K)\Big)$
Exhaustive Search	$\mathcal{O}(2^{NKL})$
Sch in [12]	$\mathcal{O}\Big((N^3L)N\Big)$

Note that we are unable to calculate the optimal performance, due to the high complexity computation for a large number of BSs, UEs, and pilots. Specifically, with $K = 4, N = 20$, and $L = 50$, one independent realization has 2^{4000} possible combinations.

A. COMPLEXITY AND CONVERGENCE

Figure 3 shows that the network spectral efficiency performance of Alg. 1 and 2 differ by on average 5%, albeit the iteration requirement of Alg. 2 is 100 times less than Alg. 1. On average, Alg. 1 needs 3×10^4 iterations and Alg. 2 needs 79 iterations. Alg. 2 has a lower number of iterations because it reduces the number of interacting actions and agents and uses a heuristic method to associate UEs with BSs. By reducing the number of interacting actions and agents, the optimization tasks can be significantly decreased. In addition, by incorporating a heuristic method in the multi-BS association, Alg. 2 practically only does the pilot allocation in the learning algorithm compared to Alg. 1, which allocates

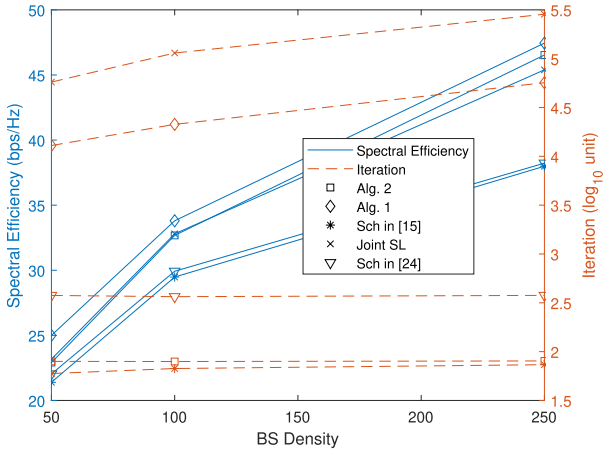


FIGURE 3. Spectral efficiency and iteration comparison with $K = 4$ for several schemes.

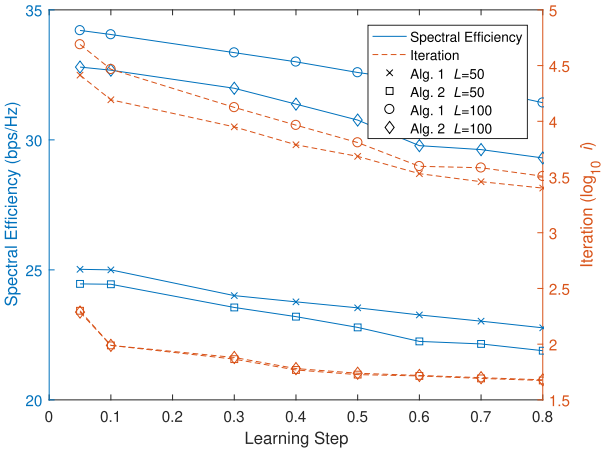


FIGURE 4. Spectral efficiency and iteration comparison for different learning step (θ), $K = 4$.

pilots and associates BSs during the learning process. As a result, Alg. 1 needs more iterations to converge. It can also be observed that Joint SL has the largest number of iterations among the schemes with 1.5×10^5 iterations on average. In addition, Sch in [15] has the lowest number of iterations, albeit a lower network spectral efficiency compared to Alg. 2, and Sch in [24] has the lowest spectral efficiency with a relatively higher number of iterations compared to the number of iterations in Alg. 2. On average, Sch in [15] needs 66 iterations and sch in [24] needs 370 iterations. A further analysis of spectral efficiency performance is discussed in Subsection V-D. To conclude, we can improve the spectral efficiency by solving the pilot allocation and multi-BS association and further reduce the number of iterations by incorporating a heuristic solution in the multi-BS association.

We calculate the complexity of the simulated schemes and the exhaustive search in Table 2. The complexity in Table 2 is defined as the computational cost of an algorithm in terms of the combinations that need to be searched. In Alg. 1, to solve (20) we decompose the optimization function into NL agents. Each agent has to update $(K + 1)$ actions in parallel. Thus, the total complexity is $(K + 1)i_1$, where i_1 is the number of iterations required for the action probability to converge according to Line 10 of Alg. 1. Alg. 2 decomposes the optimization function to N agents and has to update K actions. The additional multi-BS association by the heuristic method yields a complexity of NL. Thus, the total complexity is $(NL+K)i_2$, where i_2 is the number of iterations in Alg. 2. We also compare the proposed scheme with other schemes. Table 2 shows that the total complexity of Joint SL is the highest when we include i_4, which denotes the number of iterations in Joint SL. The total complexity of Joint SL and Alg. 1 are the highest among the algorithms. In general, if we incorporate a heuristic solution in the multi-BS association based on the long-term channel gain strength, the total complexity will be lower as done in Sch in [15], [24], and Alg. 2. In addition, we can calculate the complexity of Sch in [24] with i_5 that denotes the number of its iterations and Sch in [15] with i_3 that denotes the number of its iterations. It shows that the parallel scheme in [24] has a lower total complexity compared to the scheme in [15]. Then, we can directly compare Sch in [15] and [24] with Alg. 2. By including the average number of required iterations as stated in the previous paragraph, we can say the complexity of Alg. 2 is approximately $\frac{1}{NK}$ times that of Sch in [15] and $\frac{NL}{5K}$ of Sch in [24]. By also observing the network spectral efficiency result in Fig. 3 and given K, L, N, we can conclude that the proposed scheme in Alg. 2 maintains a high network spectral efficiency with a relatively low complexity.

B. LEARNING STEP

Figure 4 shows the effect of a variable learning step θ on the downlink network spectral efficiency and the number of iterations by using Alg. 1 and Alg. 2, respectively. The y-axis shows the network spectral efficiency and the number of iterations represented in $\log_{10}$ scale and the x-Axis shows the learning step. The figure shows that by decreasing θ, we can improve the network spectral efficiency at the cost of having an increase in the number of iterations. This happens because by decreasing θ, the rate needed by the action probability to converge to 1 is slower. Therefore, the agents can explore other available actions, which might increase the network spectral efficiency. Thus, we select the value of θ based on the trade-off between the network spectral efficiency and the number of iterations. Fig. 4 shows that the network spectral efficiency only increases slightly when the learning step decreases from 0.1 to 0.05. However, the network spectral efficiency improvement at the learning step of 0.05 requires almost twice the number of iterations at the learning step of 0.1. Therefore, we choose the value of θ to be 0.1 for the remaining numerical simulations in this section.

C. UE OUTAGE

Table 3 shows the percentage of UEs that do not get pilots (e.g., UE outage) over 300 independent realizations for various BS densities with $K = 2$, 4, 10 and $N = 20$. The values

TABLE 3. UE Outage for different number of BSs (L), number of pilots (K), and $N = 20$.

Number of Pilots (K)	Scheme	Number of BSs (L)		
		50	100	250
2	Alg. 1	0	0	0
	Alg. 2	0	0	0
	Sch. in [15]	0.2912	0.2425	0.195
	Sch. in [24]	0.285	0.2302	0.1858
	Joint SL	0	0	0
	Sch in [12]	0.9	0.9	0.9
4	Alg. 1	0	0	0
	Alg. 2	0	0	0
	Sch. in [15]	0.1058	0.034	0.00583
	Sch. in [24]	0.0863	0.0273	0.005167
	Joint SL	0	0	0
	Sch in [12]	0.8	0.8	0.8
10	Alg. 1	0	0	0
	Alg. 2	0	0	0
	Sch. in [15]	0	0	0
	Sch. in [24]	0	0	0
	Joint SL	0	0	0
	Sch in [12]	0.5	0.5	0.5
20	Alg. 1	0	0	0
	Alg. 2	0	0	0
	Sch. in [15]	0	0	0
	Sch. in [24]	0	0	0
	Joint SL	0	0	0
	Sch in [12]	0	0	0

in the third to fifth column represent the average number of outages divided by the number of UEs in the network, $N = 20$. When the number of pilots and the number of BSs in Sch in [24] and Sch [15] for a given K increases, UE outage percentage is decreased. Our proposed Alg. 1 and Alg. 2 and the Joint SL show no outage for any number of pilots. This is because we jointly solve the multi- BS association and pilot allocation. In the joint optimization, the solution of pilot allocation and multi-BS association is obtained together in each iteration. If there is a solution that leaves some UEs without pilots (UE outage), we keep searching for the solution, which does not cause a UE outage. Note that, when the number of BSs is larger than the number of UEs, all UEs must be able to get pilots. In contrast, the schemes in [15] and in [24] have at least one UE that does not get a pilot when $K = 2, 4$ for every number of BSs. It can also be observed that the percentage of UE outages for Sch in [15] is slightly higher compared to Sch in [24], due to the sequential optimization taking turn per UE. In the sequential optimization, the next optimized UE may not get pilots because there are limited pilots available from the associated BSs; the previously optimized UEs may select pilots that cause the next optimized UE to be unable to get any pilot. Sch in [24] has a slightly lower UE outage because it optimizes the pilot allocation in parallel. In the parallel optimization, since the pilots for UEs are optimized at the same time, the possible solutions that may cause a higher UE outage can be avoided. Nevertheless, a separate optimization always allows the possibility of a UE outage. In the separate process, multi-BS association is done prior to the pilot allocation. In the multi-BS association, UEs may share the same BSs. After the pilot allocation, the UE outage happens because the number of pilots at the shared BSs is fewer than the number of UEs.

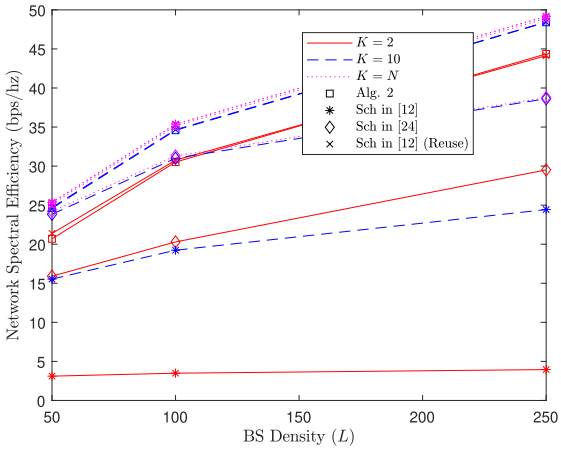


FIGURE 5. Network spectral efficiency comparison of different schemes for $K = 2$, 10, 20 and $N = 20$.

D. NETWORK SPECTRAL EFFICIENCY

In this subsection, we discuss and compare the network spectral efficiency performance of the schemes in the literature and our proposed algorithm for different numbers of pilots and BS density as shown in Fig. 5. We exclude some schemes in both joint optimization and separate optimization. For the joint optimization scheme, we exclude Joint SL and Alg. 1 because their spectral efficiency is only approximately 5% better than Alg. 2 but with higher complexity, as shown in Fig. 3 and Table 2. For the separate optimization scheme, we exclude Sch in [15] because it has a similar spectral efficiency with Sch in [24] but with much higher complexity. Note that, for $K = N$, Sch in [24] does not have PC interference and only associates BSs to UEs according to (27), which determines the multi-BSs association rule. In Fig. 5, we can observe that in general the spectral efficiency increases with the growing BS density and the number of pilots. Moreover, the network spectral efficiency performance of Alg. 2 only increases by 4.8839 or 15.13% on average from $K = 2$ to $K = N$. This shows that our joint proposed scheme utilizes multi-BS association and pilot allocation to mitigate PC effectively even in the severe PC interference case such as $K = 2$. This is unlike the separate optimization in Sch in [24] that has a larger gap between $K = 2$ and $K = N$, which is 30% on average. For a small number of pilots, i.e. $K = 2$, our joint proposed scheme outperforms Sch in [24] by 31.01% on average. For the higher number of pilots, i.e. $K = 10, 20$, our joint proposed scheme outperforms the separate optimization by 17.45%. For a small number of pilots, the spectral efficiency performance of Sch in [24] is highly affected by the UE outage resulting in fewer than N UEs that can be served. In addition, the performance of separate optimization also depends highly on the value of $\delta\%$ in (27), which explains the multi-BS association rule. Note that, $\delta\%$ represents the threshold ratio between the sum of the long-term fading of the associated BSs to a UE and the sum of the long-term fading of all BSs to a UE. For instance, $\delta = 90$ means that the sum of the long-term fading of a UE

link to the associated BSs must be at least 90% of the total long-term fading of a UE link to all BSs. For $\delta = 90$, a UE is served by all BSs. Thus, it affects the SINR value of a UE. In this case, the value of the desired signal is determined by the associated BSs and the interference comes from the BSs that serve other UEs. Moreover, δ is obtained by trial and error and it has different results for different configurations of BSs and UEs.

E. PROPOSED SCHEME VS SCH IN [12]

In this subsection, we compare the performance of Alg. 2 and Scheme (Sch) in [12]. Sch in [12] employs a cell-free massive MIMO approach, where a UE must be associated with all BSs and does a pilot allocation optimization to UEs. Table 2 shows that the computational complexity of Sch in [12] depends on the number of UEs (N) and the number of BSs (L). Note that, the number of iterations used in the "Tabu Search" is set as N, following [12], resulting in computational complexity of $(N^3L)N$. From Table 2, it could be seen that the complexity of Alg. 2 is approximately $\frac{i_2}{N^3}$ of Sch in [12]. Note that from Fig. 3, we could conclude that $i_2 = 79.64 \approx 80$ is the average number of iterations needed by one action probability of agents in Alg. 2 to converge to unity, e.g. 0.99, for $K = 4$, $N = 20$, and $L = 50{,}100{,}250$. Thus, by plugging i_2 and N in $\frac{i_2}{N^3}$, the computational complexity of Alg. 2 is approximately 100 times lower than the one in [12] for this setup. Fig. 5 shows that the network spectral efficiency performance of Sch in [12] is much lower than Alg. 2 when $K < N$ and they have similar performance when $K = N$. This is because in a cell-free MIMO system, there is only a single virtual-cell, where all BSs are associated with K UEs. As a result, the same pilot cannot be reused for different UEs [13], leading to the UE outage. In contrast, in Alg. 2, where multiple virtual cells are formed, pilots can be reused, preventing UE outage, regardless of the number of pilots. In Alg. 2, when the number of associated UEs at a BS is equal to the number of pilots, UEs can be associated with other BSs. Note that, in the case of distributed massive MIMO, where the number of BSs is larger than the number of UEs, all UEs must be able to get pilots from BSs. Table 3 validates the above argument in which the UE outage in Sch in [12] is shown. To conclude, our proposed scheme yields a better network spectral efficiency with lower complexity as compared to Sch in [12].

During the review process, one of the reviewers mentioned that in a cell-free massive MIMO where all antennas act a single virtual BS, the same pilot can be reused for different UEs. We have simulated Sch in [12] under this setting in Fig. 5, denoted as Sch in [12] (reuse). Note that although Alg. 2 has a similar network spectral efficiency performance with Sch in [12] (reuse), it has 100 times lower computational complexity. This is shown in Table 2. In reality and also in real systems, as stated in [13], pilots are used to identify UEs. Thus, if the same pilots are used by UEs within a single virtual BS, the BS will not be able to uniquely identify the UEs.

VI. CONCLUSION

In this paper, we have developed a pursuit learning-based joint pilot allocation and multi-BS association. We present a theoretical proof and numerical simulations showing that the pursuit learning algorithm converges to unity. Furthermore, we evaluate the performance of the proposed scheme by simulation. First, we show that by including a heuristic solution in the pursuit learning algorithm, we can reduce the number of actions to lower the complexity. Second, we show that the learning step size can affect both the spectral efficiency and the number of iterations required for the algorithm to converge, and it can be selected to achieve a favorable trade-off between the two parameters. Third, we show that by using a joint optimization of the pilot allocation and multi-BS association, we can maintain a high network spectral efficiency without causing UE outage.

APPENDIXES

APPENDIX A
THE DERIVATION OF (14) OR E$\{|S_A|^2\}$

The MMSE channel estimate $\hat{h}_{n,l}$ consists of the actual channel and the channel estimation error, i.e. $e_{n,l} = h_{n,l} - \hat{h}_{n,l}$. By substituting $h_{n,l} = e_{n,l} + \hat{h}_{n,l}$ into S_A in (12), we have

$$\begin{aligned} S_A &= \sqrt{\rho_p}\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\mathrm{E}\{\hat{h}_{n,l}^H h_{n,l} v_{n,l}^k\}q_n \\ &= \sqrt{\rho_p}\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\mathrm{E}\{\hat{h}_{n,l}^H (e_{n,l}+\hat{h}_{n,l}) v_{n,l}^k\}q_n \\ &= \sqrt{\rho_p}\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\beta_{n,l}\kappa_{n,l}^k v_{n,l}^k q_n \end{aligned} \tag{45}$$

Thus, E$\{|S_A|^2\}$ is given as

$$\mathrm{E}\{|S_A|^2\} = \rho_p\Big(\sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{n,l}^k}\beta_{n,l}\kappa_{n,l}^k v_{n,l}^k\Big)^2 \tag{46}$$

APPENDIX B
THE DERIVATION OF (15) OR E$\{|S_B|^2\}$

S_B can be written as follows

$$S_B = \sum_{\substack{n=1 \\ \acute{n}\neq n}}^{N} S_{\tilde{B}_{n\acute{n}}} q_{\acute{n}} \tag{47}$$

where $S_{\tilde{B}_{n\acute{n}}}$ is defined as follows,

$$\begin{aligned} S_{\tilde{B}_{n\acute{n}}} &= \sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{\acute{n},l}^k}\hat{h}_{\acute{n},l}^H h_{n,l} v_{\acute{n},l}^k \\ &= \sum_{k=1}^{K}\sum_{l=1}^{L}\sqrt{\alpha_{\acute{n},l}^k}\kappa_{\acute{n},l}^k\Big(\sqrt{\rho_p}h_{n,l}v_{\acute{n},l}^k\boldsymbol{\varphi}_{\acute{n}}^H\boldsymbol{\varphi}_n \\ &\quad + \sqrt{\rho_p}\sum_{\substack{\acute{\imath}=1 \\ \acute{n}\neq n}}^{N}\sum_{\substack{\acute{l}=1 \\ \acute{l}\neq l}}^{L} h_{\acute{\imath},l} v_{\acute{n},\acute{l}}^k \boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_{\acute{\imath}} + \tilde{z}_l\Big)^H h_{n,l} v_{\acute{n},l} \end{aligned} \tag{48}$$

Then, we take the expectation of the square value of $S_{\tilde{B}_{n\acute{n}}}$, i.e. $\mathrm{E}\{|S_{\tilde{B}_{n\acute{n}}}|^2\}$, as shown in (49) at the bottom of this page. We can proceed to compute $S_{\tilde{B}_{1_{n\acute{n}}}}$, $S_{\tilde{B}_{2_{n\acute{n}}}}$, and $S_{\tilde{B}_{3_{n\acute{n}}}}$ as follows

$$S_{\tilde{B}_{1_{n\acute{n}}}} = \rho_p |\boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_n|^2 \left(\sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{\alpha_{\acute{n},l}^k} \kappa_{\acute{n},l}^k \beta_{n,l} v_{\acute{n},l}^k \right)^2 \tag{50}$$

$$S_{\tilde{B}_{2_{n\acute{n}}}} = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N} \sum_{\substack{\acute{l}=1 \\ \acute{l}\neq l}}^{L} \alpha_{\acute{n},l}^k \rho_p (\kappa_{\acute{n},l}^k)^2 \beta_{\acute{n},l} \beta_{n,l} v_{\acute{n},\acute{l}}^k |\boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_{\acute{n}}|^2 \tag{51}$$

$$S_{\tilde{B}_{3_{n\acute{n}}}} = \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{\acute{n},l}^k (\kappa_{\acute{n},l}^k)^2 \beta_{n,l} v_{\acute{n},l}^k \sigma_{z_l}^2 \tag{52}$$

$S_{\tilde{B}_{2_{n\acute{n}}}}$ and $S_{\tilde{B}_{3_{n\acute{n}}}}$ in (51) and (52) can be combined as

$$S_{\check{B}_{n\acute{n}}} = \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{\acute{n},l}^k \beta_{n,l} (\kappa_{\acute{n},l}^k)^2 \left(\rho_p \sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N} \sum_{\substack{\acute{l}=1 \\ \acute{l}\neq l}}^{L} \beta_{\acute{n},l} v_{\acute{n},\acute{l}}^k \times |\boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_{\acute{n}}|^2 + v_{\acute{n},l}^k \sigma_{z_l}^2 \right) \tag{53}$$

By solving (53), we can obtain

$$S_{\check{B}_{n\acute{n}}} = \sqrt{\rho_p} \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{\acute{n},l}^k \kappa_{\acute{n},l}^k \beta_{\acute{n},l} \beta_{n,l} v_{\acute{n},l}^k \tag{54}$$

Thus $\mathrm{E}\{|S_B|^2\}$ is given as

$$\mathrm{E}\{|S_B|^2\} = \rho_p \sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N} |\boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_n|^2 \left(\sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{\alpha_{\acute{n},l}^k} \kappa_{\acute{n},l}^k \beta_{n,l} v_{\acute{n},l}^k \right)^2 + \sqrt{\rho_p} \sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N} \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{\acute{n},l}^k \kappa_{\acute{n},l}^k \beta_{\acute{n},l} \beta_{n,l} v_{\acute{n},l}^k \tag{55}$$

APPENDIX C
THE DERIVATION OF (16) OR E{|S_C|^2}

$\mathrm{E}\{|S_C|^2\}$ is given as follows

$$\begin{aligned}
&\mathrm{E}\{|S_C|^2\} \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{n,l}^k \mathrm{E}\{|\hat{h}_{n,l}^H h_{n,l} v_{n,l}^k - \mathrm{E}\{\hat{h}_{n,l}^H h_{n,l} v_{n,l}^k\}|^2\} \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{n,l}^k \left(\mathrm{E}\{|\hat{h}_{n,l}^H h_{n,l} v_{n,l}^k|^2\} - |\mathrm{E}\{\hat{h}_{n,l}^H h_{n,l} v_{n,l}^k\}|^2 \right) \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{n,l}^k \Big(\mathrm{E}\{|(|\hat{h}_{n,l}|^2 + e_{n,l} \hat{h}_{n,l}^H) v_{n,l}^k|^2\} \\
&\quad - \left[\sqrt{\rho_p} \beta_{n,l} \kappa_{n,l} v_{n,l}^k \right]^2 \Big) \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{n,l}^k \Big(v_{n,l}^k \left[\mathrm{E}\{|\hat{h}_{n,l}|^4\} + \mathrm{E}\{|e_{n,l} \hat{h}_{n,l}|^2\} \right] \\
&\quad - \left[\sqrt{\rho_p} \beta_{n,l} \kappa_{n,l}^k v_{n,l}^k \right]^2 \Big) \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{n,l}^k v_{n,l}^k \Big(2 \left[\sqrt{\rho_p} \beta_{n,l} \kappa_{n,l}^k \right]^2 + \sqrt{\rho_p} \beta_{n,l} \kappa_{n,l}^k \\
&\quad \times (\beta_{n,l} - \sqrt{\rho_p} \beta_{n,l} \kappa_{n,l}^k) - \left[\sqrt{\rho_p} \beta_{n,l} \kappa_{n,l}^k \right]^2 \Big) \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_{n,l}^k v_{n,l}^k \left(\sqrt{\rho_p} \beta_{n,l}^2 \kappa_{n,l}^k \right)
\end{aligned} \tag{56}$$

ACKNOWLEDGMENT

This article was presented in part at the IEEE Wireless Communications and Networking Conference, Seoul, South Korea, 2020.

$$\begin{aligned}
\mathrm{E}\{|S_{\tilde{B}_{n\acute{n}}}|^2\} &= \mathrm{E}\Bigg\{ \Bigg| \sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{\alpha_{\acute{n},l}^k} \sqrt{\rho_p} \kappa_{\acute{n},l}^k |h_{n,l}|^2 v_{\acute{n},l}^k \boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_n + \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N} \sum_{\substack{\acute{l}=1 \\ \acute{l}\neq l}}^{L} \sqrt{\alpha_{\acute{n},l}^k} \sqrt{\rho_p} \kappa_{\acute{n},l}^k h_{\acute{n},l}^H h_{n,l} v_{\acute{n},\acute{l}}^k \boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_{\acute{n}} \\
&\quad + \sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{\alpha_{\acute{n},l}^k} \kappa_{\acute{n},l}^k \tilde{z}_l^H h_{n,l} v_{\acute{n},l}^k \Bigg|^2 \Bigg\} \\
&= \underbrace{\mathrm{E}\{|\sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{\alpha_{\acute{n},l}^k} \sqrt{\rho_p} \kappa_{\acute{n},l}^k |h_{n,l}|^2 v_{\acute{n},l}^k \boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_n|^2\}}_{S_{\tilde{B}_{1_{n\acute{n}}}}} + \\
&\quad \underbrace{\mathrm{E}\{|\sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{\substack{\acute{n}=1 \\ \acute{n}\neq n}}^{N} \sum_{\substack{\acute{l}=1 \\ \acute{l}\neq l}}^{L} \sqrt{\alpha_{\acute{n},l}^k} \sqrt{\rho_p} \kappa_{\acute{n},l}^k h_{\acute{n},l}^H h_{n,l} v_{\acute{n},\acute{l}}^k \boldsymbol{\varphi}_{\acute{n}}^H \boldsymbol{\varphi}_{\acute{n}}|^2\}}_{S_{\tilde{B}_{2_{n\acute{n}}}}} + \underbrace{\mathrm{E}\{|\sum_{k=1}^{K} \sum_{l=1}^{L} \sqrt{\alpha_{\acute{n},l}^k} \kappa_{\acute{n},l}^k \tilde{z}_l^H h_{n,l} v_{\acute{n},l}^k|^2\}}_{S_{\tilde{B}_{3_{n\acute{n}}}}}
\end{aligned} \tag{49}$$

REFERENCES

[1] H. Moon, "Channel-adaptive random access with discontinuous channel measurements," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1704–1712, May 2016.
[2] S. Chen, S. Sun, Y. Wang, G. Xiao, and R. Tamrakar, "A comprehensive survey of TDD-based mobile communication systems from TD-SCDMA 3G to TD-LTE(A) 4G and 5G directions," *China Commun.*, vol. 12, no. 2, pp. 40–60, Feb. 2015.
[3] 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.
[4] R. K. Mungara, G. Caire, O. Y. Bursalioglu, C. Wang, and H. C. Papadopoulos, "Fog massive MIMO with on-the-fly pilot contamination control," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 41–45.
[5] J. Zuo, J. Zhang, C. Yuen, W. Jiang, and W. Luo, "Energy-efficient downlink transmission for multicell massive DAS with pilot contamination," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1209–1221, Feb. 2017.
[6] C. Pan, H. Mehrpouyan, Y. Liu, M. Elkashlan, and N. Arumugam, "Joint pilot allocation and robust transmission design for ultra-dense user-centric TDD C-RAN with imperfect CSI," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2038–2053, Mar. 2018.
[7] X. Zhu, Z. Wang, C. Qian, L. Dai, J. Chen, S. Chen, and L. Hanzo, "Soft pilot reuse and multicell block diagonalization precoding for massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3285–3298, May 2016.
[8] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.
[9] 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.
[10] R. Mochaourab, E. Bjornson, and M. Bengtsson, "Adaptive pilot clustering in heterogeneous massive MIMO networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5555–5568, Aug. 2016.
[11] 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.
[12] H. Liu, J. Zhang, X. Zhang, A. Kurniawan, T. Juhana, and B. Ai, "Tabu-search based pilot assignment for cell-free massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2286–2290, Feb. 2019.
[13] O. Y. Bursalioglu, G. Caire, R. K. Mungara, H. C. Papadopoulos, and C. Wang, "Fog massive MIMO: A user-centric seamless hot-spot architecture," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 559–574, Jan. 2019.
[14] Z. Chen, X. Hou, and C. Yang, "Training resource allocation for user-centric base station cooperation networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2729–2735, Apr. 2016.
[15] Y. Lin, R. Zhang, C. Li, L. Yang, and L. Hanzo, "Graph-based joint user-centric overlapped clustering and resource allocation in ultradense networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4440–4453, May 2018.
[16] J. Zhang, X. Yuan, and Y. J. Zhang, "Locally orthogonal training design for cloud-RANs based on graph coloring," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6426–6437, Oct. 2017.
[17] C. Pan, H. Ren, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Weighted sum-rate maximization for the ultra-dense user-centric TDD C-RAN downlink relying on imperfect CSI," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1182–1198, Feb. 2019.
[18] P. Valente Klaine, M. Jaber, R. D. Souza, and M. A. Imran, "Backhaul aware user-specific cell association using Q-Learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3528–3541, Jul. 2019.
[19] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2217–2228, Aug. 2017.
[20] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui, "Load balancing for ultra-dense networks: A deep reinforcement learning based approach," 2019, *arXiv:1906.00767*. [Online]. Available: http://arxiv.org/abs/1906.00767
[21] J. Xu, P. Zhu, J. Li, and X. You, "Deep learning-based pilot design for multi-user distributed massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1016–1019, Aug. 2019.
[22] K. Kim, J. Lee, and J. Choi, "Deep learning based pilot allocation scheme (DL-PAS) for 5G massive MIMO system," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 828–831, Apr. 2018.
[23] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.
[24] Y. Sun, M. Peng, and H. V. Poor, "A distributed approach to improving spectral efficiency in uplink device-to-device-enabled cloud radio access networks," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6511–6526, Dec. 2018.
[25] Y. Xu, J. Wang, Q. Wu, J. Zheng, L. Shen, and A. Anpalagan, "Dynamic spectrum access in time-varying environment: Distributed learning beyond expectation optimization," *IEEE Trans. Commun.*, vol. 65, no. 12, pp. 5305–5318, Dec. 2017.
[26] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, Jul. 2013.
[27] P. Cheng, C. Ma, M. Ding, Y. Hu, Z. Lin, Y. Li, and B. Vucetic, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.
[28] T. Marzetta, E. Larsson, H. Yang, and H. Ngo, *Fundamentals of Massive MIMO*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[29] *Spatial Channel Model for Multiple Input Multiple Output (MIMO) Simulations*, document 3GPP Std. TR 25.996 v. 12.0.0, Third Generation Partnership Project (3GPP), Sep. 2014.
[30] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[31] J. Hoydis, S. ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.
[32] G. Caire, "On the ergodic rate lower bounds with applications to massive MIMO," 2017, *arXiv:1705.03577*. [Online]. Available: http://arxiv.org/abs/1705.03577
[33] 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.
[34] Y.-G. Lim, C.-B. Chae, and G. Caire, "Performance analysis of massive MIMO for cell-boundary users," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6827–6842, Dec. 2015.

NAUFAN RAHARYA (Student Member, IEEE) received the B.S. and M.S. degrees from the University of Indonesia, in 2014 and 2015, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Information Engineering, The University of Sydney. His research interests include ultradense networks, user association, resource allocation, massive MIMO, and machine learning application for wireless communications. He was a recipient of postgraduate scholarships from Indonesia Endowment Fund for Education (LPDP) by the Government of Indonesia.

WIBOWO HARDJAWANA (Member, IEEE) received the Ph.D. degree in electrical engineering from The University of Sydney, Australia, in 2009. From 1999 to 2004, he was with Singapore Telecom Ltd. He is currently an Australian Research Council Discovery Early Career Research Award Fellow with the School of Electrical and Information Engineering, The University of Sydney. His current research interests include software defined wireless cellular networks, with a focus on radio resource allocation algorithms, MIMO, network architecture, and prototype development.

OBADA AL-KHATIB (Member, IEEE) received the B.Sc. degree (Hons.) in electrical engineering from Qatar University, Qatar, in 2006, the M.Eng. degree (Hons.) in communication and computer from the National University of Malaysia, Bangi, Malaysia, in 2010, and the Ph.D. degree in electrical and information engineering from The University of Sydney, Sydney, Australia, in 2015. From 2006 to 2009, he was an Electrical Engineer with Consolidated Contractors International Company, Qatar. In 2015, he joined the Centre for IoT and Telecommunications, The University of Sydney, as a Research Associate. Since 2016, he has been an Assistant Professor with the Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai, United Arab Emirates. His current research interests include the areas of smart grid communication, wireless resource allocation and management, cooperative communications, and wireless network virtualization.

BRANKA VUCETIC (Life Fellow, IEEE) is currently an ARC Laureate Fellow and the Director of the Centre of Excellence for IoT and Telecommunications, The University of Sydney. Her current work are in the areas of wireless networks and the Internet of Things. In the area of wireless networks, she explores possibilities of millimetre wave (mmWave) frequency bands. In the area of the Internet of Things, she works on providing wireless connectivity for mission critical applications. She is also a Fellow of the Australian Academy of Technological Sciences and Engineering and the Australian Academy of Science.