Self-Imitation Learning based Inter-Cell Interference Coordination in Autonomous HetNets

Mu Yan*, Jian Yang*, Keyu Chen†, Yao Sun†, Gang Feng†, *Senior Member, IEEE*
* Northern Institute of Electronic Equipment of China, Beijing 100191, China
† University of Electronic Science and Technology of China, Chengdu, China
Email: nanwuyaoshi@163.com

***Abstract*—Recently, mobile operators have been shifting to an intelligent autonomous network paradigm, where the mobile networks are automated in a plug-and-play manner to reduce the manual intervention. Under this circumstance, serious inter-cell interference becomes inevitable which may severely deteriorate system throughput performance and users' quality of service (QoS), especially for dense residential small base station (SBS) deployment. This paper proposes an intelligent inter-cell interference coordination (ICIC) scheme for autonomous heterogeneous networks (HetNets), where the SBSs agilely schedule sub-channels to individual users at each Transmit Time Interval (TTI) with aim of mitigating interferences and maximizing long-term throughput by sensing the environment. Since the reward function is inexplicit and only few samples can be used for prior-training, we formulate the ICIC problem as a distributed inverse reinforcement learning (IRL) problem following the POMDP games. We propose a non-prior knowledge based self-imitating learning (SIL) algorithm which incorporates Wasserstein Generative Adversarial Networks (WGANs) and Double Deep Q Network (Double DQN) algorithms for performing behavior imitation and few-shot learning in solving the IRL problem from both the *policy* and *value*. Numerical results reveal that SIL is able to implement TTI level's decision-making to solve the ICIC problem, and the overall network throughput of SIL can be improved by up to 19.8% when compared with other known benchmark algorithms.**

***Index Terms*—Autonomous HetNets, Inter-Cell Interference Coordination, POMDP, Wasserstein GANs, Double DQN.**

I. INTRODUCTION

According to the Cisco Visual Networking Index [1], mobile traffic has occupied a large portion of the big datasets, and more than 70% of data traffic and 50% of voice calls occur indoors. Mobile networks or cellular networks aiming at improving the performance of the cell-edge users have been impeded by the ever-increasing traffic demand. Driven by the explosive growth of traffic demand, dense deployment of HetNets which combines various cellular networks and massive plug-and-play small-cell base stations (SBSs) have been introduced as an effective architectural technology for improving the spatial use and network capacity of 5G-and-beyond cellular networks [2]–[6].

Dense deployment of plug-and-play SBSs may lead to severe inter-cell interference (ICI), which significantly deteriorates both the network throughput and the quality of service (QoS) of users. Thus inter-cell interference coordination (ICIC) is of vital importance for indoor coverage of mobile communication systems. Recent researches aiming at solving the ICIC problem for HetNets mainly focus on the techniques of power allocation and subcarriers scheduling for interference mitigation [7], [8]. While ICIC in HetNets has been extensively studied [8]–[13], the research in decentralized HetNets, where the SBSs make resource allocation decisions independently without information sharing, is still far from adequate, especially for the case of indoor deployment of plug-and-play SBSs. In recent investigations [10]–[12], the authors consider that the overall network information can be assembled via a central controller, and the global optimum of resource management for SBSs can be accomplished with the central controller. However, frequent information exchange between SBSs and decision making of central controller for large-scale SBSs are very costly [14]. Especially when SBSs are managed by different mobile operators, exchanging information between SBSs and central control are even infeasible. In this case, the SBSs can only perform decentralized resource scheduling by exploiting partially observable network information.

In autonomous HetNets, the inter-cell interferences are usually time-varying due to the switch on/off of small plug-and-play SBSs and mobility of users. Therefore, it is ineffective to use static optimization based algorithms [7], [8] and other heuristic algorithms such as game theory [11], [15] to solve the ICIC problem in autonomous HetNets, due to the poor adaptability and generalization for the dynamic environment. This inspires us to exploit partially observable network information to schedule spectral resources of the HetNets to individual users in an adaptive and intelligent way, with aim of minimizing ICI. Fortunately, recent emerging machine learning algorithms such as deep reinforcement learning, which continuously improves strategies by timely interacting with the environment and evolves with the learning epochs, is able to provide an effective tool to address this challenging problem.

This paper investigates the ICIC in autonomous HetNets, where SBSs could be owned and operated by different Mobile Network Operators (MNOs) and cannot exchange state information due to the backhaul constraints [16]. We resort to learning algorithms with strong adaptability and evolutionary ability for performing ICIC in autonomous HetNets. Specifically, embedded with updatable neural networks, the proposed learning algorithms can evolve with the training processes. By modeling the sequential decision-making process as a Partially Observable Markov Decision Process (POMDP), individual SBSs implement resource scheduling in an autonomous manner by sensing the surrounding environment at each TTI. For a

specific SBS, the reward function used for evaluating its strategies is inexplicit and the policies of other SBSs are unknown. In addition, the number of the samples used for training the algorithm in advance is limited at the boot-up stage of SBSs. For addressing these difficulties, we propose an inverse reinforcement learning (IRL) based self-imitation learning (SIL) framework, which consists of Wasserstein Generative Adversarial Networks (W-GANs) [17] and Double Deep Q-Network (Double DQN) [18], working in a collaborative way. In more detail, there are two miniature neural networks in W-GANs: generative network model G and discriminative network model D. G is used to capture the distribution of the real dataset while D is used to estimate the probability of a sample coming from the real dataset rather than from G. The training procedure for G is to maximize the probability of D making a mistake [19]. Double DQN is improved based on the Natural DQN [20] which can significantly eliminate the overestimation problem where decisions are not accurately estimated by the evaluation network. Based on that, we use Double DQN for performing the resource scheduling of SBSs. Furthermore, we combine the dataset generated by G and the real dataset generated with the learning process and put combined dataset into the replay buffer of the Double DQN, in order to reduce the correlation between samples and the probability of over-fitting.

The main contributions of this work can be summarized as follows:

- To facilitate the ICIC in autonomous HetNets, we propose to transfer the control and responsibility from the centralized controller to individual SBSs. The autonomous control stimulates the SBSs' abilities of self-learning and self-configuring with reduced signaling interactions.
- To the best knowledge of the authors, thus far, there is no priori work which collaboratively uses W-GANs and Double DQN for solving the resource allocation problem in decentralized HetNets. Particularly, W-GANs and Double DQN work in a collaborative way for tackling the problems of inexplicit reward function and few-shot learning.
- We use the state-of-the-art W-GANs for drawing policy and generating adversarial training samples with the aim of improving the sample diversity and reducing the correlation between data samples. This can further improve the generalization ability or robustness and accelerate the convergence rate of the SIL.
- To overcome the overestimation problem existing in most discrete decision-making processes, we adopt Double DQN in SIL. This helps SIL make decisions more accurate and reasonable. Moreover, in order to cater for the plug-and-play manner of indoor SBSs, the Double DQN is initialized according to the SINR, and a nested training scheme is adopted to overcome the slow-start problem of the learning process.

The rest of the paper is organized as follows. The system model is presented in Section III. In Section IV, we model the ICIC problem as non-cooperative Markov games which are specifically formulated as the distributed inverse reinforcement learning (IRL) to be solved. Next, in Section V, we design a self-imitation learning framework for solving the IRL problem. In Section VI, we present the numerical results as well as discussions, and finally conclude the paper in Section VII.

II. Related Work

ICIC problem in HetNets has recently spurred extensive investigations from different perspectives with various design objectives. ICIC can be accomplished in either decentralized or centralized manner according to network configurations. In addition, power control and subcarriers assignment are two main techniques used in solving the ICIC. In the following, we survey the main related work in the literatures from different aspects.

Optimization of Power Allocation & Subcarrier Assignment are widely used for ICIC with the aim of mitigating interferences and thus improving network performance. Particularly, joint optimization of power allocation and subcarrier assignment are recently studied in [21]–[24], in which spectral efficiency or energy efficiency is considered as the optimization objective. However, joint power allocation and subcarriers assignment in centralized HetNets is a typical multiple choice dimension knapsack problem which is known to be NP-hard [25]–[27]. Moreover, the solution of power allocation is continuous, while the solution of subcarrier assignment is discrete, making it very hard to solve this cross-domain optimization and achieve global optimality at TTI level. Thus greedy-style heuristic algorithms are usually developed to solve the challenging combinational problem in polynomial time.

From another perspective, ICIC could be addressed in either *Centralized HetNets or Decentralized HetNets*. Most existing methods for solving ICIC [8], [21], [28] are based on centralized HetNets, where a central controller is deployed to collect global information (i.e., network information and the policies of all BSs) and make decisions towards the direction of improving the overall network performance. Since the optimal joint power allocation and subcarrier assignment problem in centralized HetNets is known to be NP-hard, the computational efficiency of centralized decision-making is substantially subject to the scale of the HetNets (i.e., the number of users and subcarriers). Most researches investigate the centralized decision-making process of ICIC in small or even a single-cell network for achieving global optimality. In addition, frequent information interactions are quite resource-consuming. On the other hand, recent studies [7], [10] focusing on ICIC in decentralized HetNets show that distributed decision-making, where local decision makers are responsible for a segment of the decision process, can effectively decrease signaling overhead and performs well in most scenarios. Distributed decision-making has attracted much attention due to the rapid improvement of computing and intelligent solutions. However, restricted by the lack of information interactions, distributed decision-making is hard to achieve the global optimal solution without knowing other agents' strategies. Moreover, distributed decision-making results in a free competitive environment where each individual agent eventually achieves a Nash equilibrium [15].

Fortunately, recently emerging machine learning technologies provide a very promising tool for intelligent decision-making in uncertain and time-varying network environments. For solving resource allocation and power allocation in radio access networks (RANs), the authors of [26], [29], [30] model the decision-making process as a Markov Decision Process (MDP) and a central controller is used for collecting global network information. Accordingly, appropriate decisions can be readily made towards the direction of improving overall network performance. In comparison, distributed decision-making is more challenging because of the limited communication capabilities caused by the back-haul constraints. Although certain performance gain can be theoretically achieved in centralized HetNets, it is usually inapplicable to large-scale wireless networks due to its high computational complexity and signaling overhead. Therefore, distributed learning for autonomous decision-making is more appropriate for large-scale wireless networks, and POMDP can be used for formulating the decision-making process [31]. Besides, distributed training in decentralized HetNets is another key problem due to the implicit reward function and non prior-knowledge based training. The recently emerging W-GANs [17] provides an effective tool for coping with the problem of few training samples, which performs the sample expansion by solving the zero-sum game between the generator and discriminator with historical samples. Moreover, the Double DQN [18] can be used for addressing the problem of inexplicit reward function, by establishing relationship between the input local network information and the output performance through iteratively training the neural network.

III. System Model

This paper focuses on the orthogonal frequency division multiple access (OFDMA) based downlink heterogeneous networks (HetNets) consisting of a set of $\mathcal{B} = \{1, \cdots, B\}$ SBSs which are operated by a set of mobile operators $\mathcal{Z} = \{1, \cdots, Z\}$, and a set of $\mathcal{U} = \{1, \cdots, U\}$ user equipments (UEs) (i.e., tablets, mobile phone, etc.). Define $\mathcal{B}_{z\in\mathcal{Z}}$ as the SBSs set belonging to operator $z \in \mathcal{Z}$ and $\mathcal{U}_{b\in\mathcal{B}}$ as the users set underlying SBS $b \in \mathcal{B}$. Let $\mathcal{T} = \{1, \cdots, T\}$ be the set of decision intervals. In order to capture the network dynamics at small time granularity, we consider to implement TTI-level's decision-making. Note that the resource block (RB) is defined as the minimum transmission spectrum unit in OFDMA systems, and let $\mathcal{K}_u = \{a_{u,1}, \cdots, a_{u,K}\}$ be the index set of RBs allocated to user u, where $a_{u,k}$ represents a indicator which equals to 1 if the kth RB is assigned to user u, and 0 otherwise.

A. Network Capacity and Power Consumption

Let $E_{b,t} = [e_{u,k}]_{|\mathcal{U}_b|\times|\mathcal{K}_u|}$ be the transmit power matrix of SBS b at time t where $e_{u,k}$ denotes the transmit power received by user u on the kth subcarrier, and let $G_{b,t} = [g_{u,k}]_{|\mathcal{U}_b|\times|\mathcal{K}_u|}$ be the channel gain matrix where $g_{u,k}$ denotes the channel gain from SBS b to user u on the kth subcarrier. Let $W_{b,t} = [w_{u,k}]_{|\mathcal{U}_b|\times|\mathcal{K}_u|}$ be the matrix of the assigned subcarriers' bandwidth where $w_{u,k}$ denotes bandwidth of subcarrier k assigned to user u. Let additive white Gaussian noise (AWGN) be denoted by matrix of $N_{b,t} = [n_{u,k}]_{|\mathcal{U}_b|\times|\mathcal{K}_u|}$ where $n_{u,k}$ denotes the AWGN on subcarrier k. Let $\Gamma_{b,t} = [\gamma_{u,k}]_{|\mathcal{U}_b|\times|\mathcal{K}_u|}$ be the matrix of SINR where $\gamma_{u,k}$ denotes the SINR of user u on subcarrier k. For considering the rate assigned to a given user u, the SINR measures the signal quality and is defined as the ratio of the received sum power of the desired signal over the sum power of the interfering signals and the background noise. Therefore, the SINR matrix of SBS b at time t is given by

$$\Gamma_{b,t} = \frac{E_{b,t} \odot G_{b,t}}{N_{b,t} + \sum_{p\in\mathcal{B}\setminus\{b\}} P_{p,t} \odot G_{p,t}}, \tag{1}$$

where the notation $\odot$ represents the Hardamard product (*i.e.,* $(A \odot B)_{i,j,k} = (A)_{i,j,k}(B)_{i,j,k}$).

Let $Y_{b,t} = [y_{u,k}]_{|\mathcal{U}_b|\times|\mathcal{K}_u|}$ be the matrix of transmission rate of SBS b at time slot t where $y_{u,k}$ is the transmission rate of user u on subcarrier k, and $I_{b,t}$ be the $|\mathcal{U}_b| \times |\mathcal{K}_u|$ matrix whose elements are all equal to 1. Shannon capacity formula can be used to describe the transmission rate. Therefore, the matrix of the transmission rate $Y_{b,t}$ is derived as

$$Y_{b,t} = W_{b,t} \odot \log_2(I_{b,t} + \Gamma_{b,t}). \tag{2}$$

B. User Association and Service Constraints

Let $a_{b,u,t}$ denote the association indicator which equals to 1 if user u is associated with BS b and 0 otherwise. Throughout this paper, we assume that each UE can only be served by one BS at any time slot, and the association rule is letting users be associated with the nearest BS. Therefore we have

$$\sum_{b\in\mathcal{B}} a_{b,u,t} = 1, \forall t \geq 0. \tag{3}$$

For a given SBS b, define the the transmission rate of user u as $y_u = \sum_{k\in\mathcal{K}_u} y_{u,k}$. Let the threshold of the required transmission rate for user u be $\check{y}_u$. Therefore, the transmission rate constraint for user u associated with SBS b at time slot t is given by

$$y_u \geq \check{y}_u, u \in \mathcal{U}_{b\in\mathcal{B}}. \tag{4}$$

For a given BS b, the total amount of RBs can be used by UE u should satisfy

$$\sum_{k\in\mathcal{K}_u} a_{u,k} \leq |\mathcal{K}_u|, u \in \mathcal{U}_{b\in\mathcal{B}}. \tag{5}$$

In addition, the total transmit power consumed by SBS b is given by $\sum_{u\in\mathcal{U}_b}\sum_{k\in K_u} e_{u,k}$, and the power constraint for SBS b is represented as

$$\sum_{u\in\mathcal{U}_b}\sum_{k\in\mathcal{K}_u} e_{u,k} \leq \hat{e}, \tag{6}$$

where $\hat{e}$ is the maximum transmit power which can be managed by each SBS. Note that power control is implemented to keep the interference from other SBSs below a certain threshold. Generally, we assume that the transmit power is initially evenly allocated to each assigned subcarrier.

Then, we let $\hat{\gamma}_u$ represent the SINR threshold of user u, and the SINR constraint for a given user u is represented as

$$\gamma_{u,k} \geq \hat{\gamma}_u. \tag{7}$$

Define a widely used utility function $l(\cdot) = \ln(\cdot)$ [32] [33]. Then the rate utility function $l(y_{u,t})$ leads to resource allocation fairness for each SBS in terms of individual user rate. Therefore, we set the objective of the long-term overall utilities of an SBS as

$$\mathbb{U}_t = \sum_{t \in \mathcal{T}} \sum_{u \in \mathcal{U}_b} l(y_{u,t}). \tag{8}$$

Finally, for improving the clarity, we summarize the notations and variables used in this paper in Table I.

TABLE I
MAIN PARAMETERS AND VARIABLES

Symbol	Description
$\mathcal{B}$	SBSs set
$\mathcal{U}$	user equipments set
$\mathcal{Z}$	mobile operators set
$\mathcal{K}$	assigned RBs set
N	white gaussian noise matrix
G	channel gain matrix
E	transmit power matrix
Y	transmission rate matrix
Γ	SINR matrix
W	the assigned subcarriers' bandwidth matrix
$\mathbb{U}$	the long-term overall utilities
$\mathcal{M}$	partially observable Markov model
X	partially observable network states
$\mathcal{X}$	partially observable network state set
$\mathcal{S}$	global observable network state set
Pr	joint policy set
$\mathcal{A}$	resource allocation strategies set
Q	state-action value
V	state value

IV. PROBLEM STATEMENT AND GAME-THEORETIC SOLUTION

This work aims at improving the long-term performance on system throughput while guaranteeing the QoS requirements (i.e., interferences and rate requirement) of UEs. In this section, we first formulate the problem of interference coordination among the non-cooperative SBSs across the time horizon as a stochastic game and then discuss the best-response solution from a game-theoretic perspective.

A. Model-Free based Partially Observable Markov Decision Process (POMDP)

Due to the backhaul constraints in the autonomous HetNets, the decision-making process of the interference coordination at SBSs is defined as a POMDP as only local network states can be observed in the autonomous HetNets. Fig. 1 shows the decision-making process of the POMDP on time axis. Specifically, the POMDP $\mathcal{M}$ is modeled as a five-tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{F}, \Pr(\mathcal{X}), \gamma)$ which are respectively elaborated as below.

- $\mathcal{X}_t$ represents the partially observable network state of SBSs. Let the observed information consist of the number of UEs $\mathcal{U}_{b,t}$, and the SINR $\Gamma_{b,t} (b \in \mathcal{B})$. Let $\mathcal{X}_{b,t} = \{\mathcal{U}_{b,t}, \Gamma_{b,t}, \mathcal{X}_{b,-t}\}$, where $X_{b,-t}$ represents the information observed and saved by SBS b before time t.
- $\mathcal{A}_t$ represents a set of actions made by SBSs at time t. In the POMDP, the network states change with the actions which is defined as the subcarrier allocation strategies in this work.
- $\mathcal{F}_t : \mathcal{X}_t \times \mathcal{A}_t \times \mathcal{X}_t \rightarrow R_t$ is a family of reward functions which maps the input action $\mathcal{A}_t$ to the output reward R_t under a deterministic observed network state $\mathcal{X}_t$. More specifically, the reward function is
- $\mathbb{P}(\mathcal{X}_t)$ represents the probability under a deterministic network state $\mathcal{X}_t$. Moreover, we use $\pi_t(\tau)$ to denote the policy distribution map over a sequence of policy trajectory τ starting from time t.
- γ ($\gamma \in (0, 1)$) denotes the discount factor in the Markov chain. Discount factors are important in infinite-horizon MDPs, in which they determine how the reward is counted.

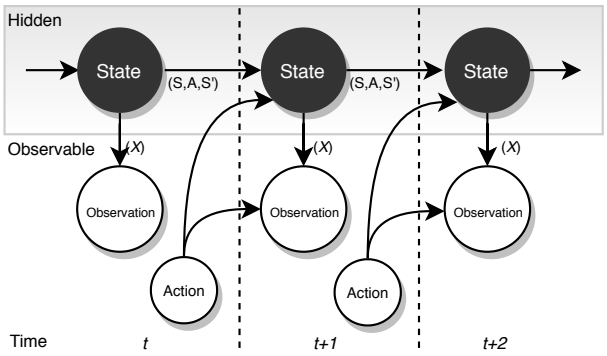


Fig. 1. Partially Observable Markov Decision Process

For a given BS b at time slot t, define $V(\mathcal{X})$ as the long-term discounted throughput starting from the initial network state X. We formulate $V(\mathcal{X}, \pi)$ which follows policy trajectory τ as a Bellman equation. For a given BS b at time slot t, $V_{b,t}(\mathcal{X}, \pi)$ is given by

$$V_{b,t}(\mathcal{X}, \pi) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{b,t+k+1} | \mathcal{X}^1 = \mathcal{X}_{t+1}], \tag{9}$$

where the expectation $\mathbb{E}(\cdot)$ is taken over decisions under different local network states following the policy trajectory π across discrete decision epochs. Moreover, the reward function is defined as $R_{b,t+1} = \sum_{u \in \mathcal{U}_b} l(r_{u,t})$. Therefore, the objective of (8) is rewritten as

$$\mathbb{U}_t(\mathcal{X}, \pi) := V_{b,t}(\mathcal{X}, \pi). \tag{10}$$

B. Stochastic Game Formulation

Define $\mathcal{S}$ as the global network states, and there exists $\mathcal{X} \subset \mathcal{S}$. Obviously, from a global optimization perspective, the objective of the POMDP is to find policy trajectory $\pi_t(\tau)$ of SBSs which can minimize the absolute value of the error between local optimum and global optimum. Based on least squares method, the objective can be formulated as

$$\pi_t^*(\tau) = \arg\min_{\pi_t(\tau)} |\mathbb{U}_t(\mathcal{X}_t, \pi) - \mathbb{U}_t(\mathcal{S}_t, \pi^*)|^2, \tag{11}$$

where $\mathbb{U}_t(\mathcal{S}_t, \pi^*)$ denotes the long-term network throughput derived by the best-response strategies.

By modeling the decision-making process as a POMDP, the HetNet system performs as an open loop system where feedbacks (i.e., strategies and states) from other BSs are impossible to be known. This implies that the reward function cannot be expressed explicitly because of the information gap. In other words, an explicit reward function $\mathcal{F}(\cdot)$ mapping the observed local network state and the global network performance is hard to be derived with limited information. Without information sharing, it is hard to judge whether the global optimal strategies are achieved in the partially observable network. Therefore, $\mathbb{U}_t(\mathcal{X}, \pi^*(\mathcal{X})) \neq \mathbb{U}_t(\mathcal{S}, \pi^*(\mathcal{S}))$ and obviously $\mathbb{U}_t(\mathcal{X}, \pi^*(\mathcal{X})) < \mathbb{U}_t(\mathcal{S}, \pi^*(\mathcal{S}))$, which means that estimating the global optimal strategies from a partially observable state space can be arbitrarily bad. This also indicates that the global optimal strategies are unknown and hard to be achieved in the decentralized HetNets.

Remark 1. *In decentralized HetNets, estimating the global optimal strategies π^* from a partially observable state space $\mathcal{X}$ can be arbitrarily bad. We then derive the best-response solution from a multi-agent game-theoretic perspective.*

The HetNet environment is time-varying with user behaviors and channel conditions, resulting in a non-stationary and competitive network environment. In order to model the non-cooperative game between SBSs, we formulate a multi-agent non-cooperative stochastic game $\mathcal{SG} := \langle \mathcal{B}, \mathcal{M} \rangle$ following the POMDP decision model $\mathcal{M}$, where the SBSs of set $\mathcal{B}$ are defined as game players or decision-making agents. Then, we consider to solve the non-cooperative game $\mathcal{SG}$ over a Nash Equilibrium (NE) game framework [34]. In detail, the performance of resource allocation strategies among agents is optimized towards the direction of Nash Equilibrium (NE) [34].

Definition 1. *In our formulated stochastic game, $\mathcal{SG}$, an NE is a tuple of control policies $\langle \pi_i^* : i \in \mathcal{B} \rangle$, where each π^* of an SBS i is the best response to the other SBSs' π_{-i}^*, i.e.,*

$$V_i(\mathcal{X}, \pi_i^*, \pi_{-i}^*) \geq V_i(\mathcal{X}, \pi_i, \pi_{-i}^*), \forall i \in \mathcal{B}. \tag{12}$$

More specifically, NE is a state of allocation of resources from which it is impossible to reallocate so as to make any one individual achieve better performance on the throughput without making at least one individual worse off [34]. An NE describes the rational behaviors of the SBSs in a stochastic game. Any resource allocation strategy that provides an NE improvement results in a non-decreasing change in individual performance.

Theorem 1. *For a multi-agent stochastic game with expected long-term discounted payoffs, and finite actions space which is visited infinitely often, there always exists an NE in stationary control policies [35].*

C. Best-Response Approach

The main objective of $\mathcal{SG}$ is to find resource allocation strategies that can achieve NE by iteratively optimizing the State-Value $V(\mathcal{X}, \pi)$ for all SBSs. Then, the long-term best-response of SBS $i \in \mathcal{B}$ starting from partially observable network state $\mathcal{X}$ can be derived as

$$V_i^*(\mathcal{X}) = \max_{\pi_i(\mathcal{X})} \{\mathbb{E}_{\pi(\tau)}[\sum_{k=0}^{\infty} \gamma^k \mathcal{F}_{i,t+k+1}(\pi_i(\mathcal{X}), \pi_{-i}^*(\mathcal{X}))|\mathcal{X}_1 = \mathcal{X}]\}, \tag{13}$$

where $V_i^*(\mathcal{X})$ denotes the State-Value of SBS i when the agents adopt the best-response strategies set $\{\pi_i^*, \pi_{-i}^*\}$,$\forall i \in \mathcal{B}$.

Note that in order to achieve the NE in the stochastic game, all SBSs have to know the global network dynamics, which is prohibited in our non-cooperative networking environment. This paper assumes that each SBS can infer actions of other SBSs by sensing the SINR condition $\Gamma_b, b \in \mathcal{B}$ of channels. Intuitively, the SINR value of channels can indirectly reflect the actions of other SBSs.

D. Decomposition of State Value: From an Inverse Reinforcement Learning Perspective

Since decision-making is prior to result perception, the reward function is inexplicit without the global decision information. Therefore, the distributed training process without prior knowledges becomes challenging in achieving the best-response strategies. Considering this, we decompose the State-Value $V(\mathcal{X})$ from an inverse reinforcement learning (IRL) perspective, with the aim of optimizing the State-Value inversely from the perception results.

To formulate the IRL problem, we redefine the variables in (9) and associate them with trajectory τ. Let $|\mathcal{S}|$ be the number of network states, and $|\mathcal{A}|$ be the number of actions. Then, we define an $|\mathcal{S}|$-dimension vector $\mathcal{R} = [R_s]_{|\mathcal{S}|\times 1}$, an $|\mathcal{S}|$-dimension vector $\mathcal{V} = [V_s]_{|\mathcal{S}|\times 1}$ and an $|\mathcal{S}|\times|\mathcal{A}|$ sequential policy matrix $\Pi = [\pi_{a|s}]_{|\mathcal{S}|\times|\mathcal{A}|}$, where R_s and V_s respectively denote the instantaneous reward and state-value at state $s \sim \tau$, and $\pi_{a|s}$ denotes the probability of adopting action a at state s. Since deterministic state transition is adopted in the POMDP, (9) can be rewritten as $\mathcal{V} = \Pi(\mathcal{R} + \gamma\mathcal{V})$, then,

$$\mathcal{V} = (\mathcal{I} - \gamma\Pi)^{-1}\Pi\mathcal{R}, \tag{14}$$

where $\mathcal{I}$ is an identity matrix and $\mathcal{I} - \gamma\Pi$ is invertible. This is because the elements of $\mathcal{I} - \gamma\Pi$ are all in the interior of a unit circle (i.e., $\pi_{a|s} \preceq 1$), and $\gamma < 1$ and thus it has no zero eigenvalues and is not singular [36].

As previously defined, the policy distribution Π is a long-term variable to be optimized, while the reward $\mathcal{R}$ is an instantaneous outcome. The key issue to improve the resource allocation strategies is to minimize the difference between

the trained policies Π and the optimal policies Π^*, and simultaneously maximize the reward $\mathcal{R}$. Then the objective of a given SBS is derived as

$$\max_{\mathcal{R}} [\min_{\Pi}\{(\Pi^* - \Pi)(\mathcal{I} - \gamma\Pi)^{-1}\mathcal{R}\}]. \tag{15}$$

Then we formulate an IRL based ICIC problem with the aim of maximizing the long-term system throughput while meeting the conditions of the NE. Specifically, for SBS b with starting time t, the IRL based ICIC problem following the POMDP is formulated as

$$\max_{\mathcal{R}}[\min_{\Pi}\{(\Pi^*_{b,t} - \Pi_{b,t})(\mathcal{I} - \gamma\Pi_{b,t})^{-1}\mathcal{R}_{b,t}\} - \lambda\|\mathcal{R}_{b,t}\|_2], \tag{16}$$

$$\text{s.t. } (\Pi^*_{b,t} - \Pi_{b,t})(\mathcal{I} - \gamma\Pi_{b,t})^{-1}\mathcal{R}_{b,t} \succeq 0, \tag{17}$$

$$X_b \cap X_p = \varnothing, \forall p \neq b \tag{18}$$

$$\langle (3), (4), (5), (6), (7) \rangle, \tag{19}$$

where $\succeq$ represents vectorial inequality (e.g., $(1, 2, 5) \succeq (1, 1, 3)$), $\|\mathcal{R}\|_2$ represents the ℓ_2-norm of $\mathcal{R}$, and $\lambda\|\mathcal{R}\|_2$ is a weight decay-like penalty term (or regularization) used to improve the over-fitting problem, and λ is an adjustable hyper-parameter. (17) is used to guarantee that the optimal policy is not worse than other policies.

As aforementioned, an explicit reward function is hard to be derived and there are few expert trajectories (or samples) can be used for pre-training in a decentralized network scenario, resulting in that the non-prior knowledge objective of (16) cannot be solved directly. Furthermore, the ICIC problem becomes more complex when considering the joint optimization from both the Policy Π and Value V of the IRL. Therefore, we decompose the IRL problem into two sub-problems which respectively implement the behavior cloning and reward function approximation, which are explained below.

- Behavior cloning: The objective of behavior cloning is to draw the policy map by imitating the expert policy trajectories which satisfies the condition that better reward is assigned to the policy with larger probability and the worse one is assigned to other policies with smaller probabilities.
- Reward function approximation: With the RL process, the reward function is approximated by neural networks utilizing the labeled training samples, and the derived reward function is used to optimize the decision-making process of RL towards the direction of improving the Value V.

Actually, there are few training samples available at the boot-up stage of an SBS, and more training samples labeled with input network states and feedback throughputs are collected during the learning process. Therefore, the training process of expert strategies is embedded in the training process of the reward function approximation. This two subproblems are trained separately, and the updated (neural) network information is synchronized periodically.

In the next section, we propose a self-imitation learning (SIL) framework which resort to machine learning tools (i.e., W-GANs and Double DQN) to collaboratively implement the behavior cloning and zero-shot RL in a distributed manner.

V. Self-Imitation Learning Framework for Decentralized HetNets

Fig. 2 shows the framework of the self-imitation learning where W-GANs and Double DQN perform in a cooperative way for implementing the zero-shot learning and the behavior cloning to solve the IRL problem of (16) and thus improve the performance of the resource allocation underlying the autonomous HetNet. In more detail, Double DQN is used for deriving the reward function by iterations and then optimizing the discrete decision making process by maximizing the value V. In addition, the basic idea of W-GANs in SIL is learning to perform a task directly from expert strategies, without estimating the corresponding reward function. Specifically, W-GANs in SIL are used to imitate the expert strategies and generate adversarial training samples, and thus help Double DQN make decisions more robust.

A. Behavior Cloning by Wasserstein-GANs

W-GANs are improved based on the conventional GANs, which does not need to maintain a careful balance in training of the discriminator and the generator, and thus an accurate design of the network architecture either [19]. The dropping phenomenon consistently happens in GANs is also significantly reduced. One of the most attractive benefits of W-GANs is the ability to continuously estimate the Earth-Mover (EM) distance by training the discriminator to optimality.

Let the expert policy be denoted by π_e. By implementing the expert policy, the decision trajectory is improved towards the direction of maximizing the network performance with large probability. Moreover, the generated policy imitated by the proposed W-GANs algorithm is denoted as π_g. W-GANs aim to optimize the EM distance or 1-Wasserstein distance [17] between π_r and π_e, which is derived as

$$W(\pi_e, \pi_g) = \inf_{\varphi \in \Pi(\pi_e, \pi_g)} \mathbb{E}_{(x,y)\sim\varphi}[\|x - y\|], \tag{20}$$

where $\Pi(\pi_e, \pi_g)$ denotes the set of joint distributions $\varphi(x, y)$ whose marginals are respectively π_e and π_g. Intuitively, $\varphi(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions π_g into the distribution π_e. The EM distance is then the "cost" of the optimal transport plan. Since the infimum in (20) is intractable, by considering the Kantorovich-Rubinstein duality [37], (20) is transformed to

$$W(\pi_e, \pi_g) = \sup_{\|D\|_L \leq 1} \mathbb{E}_{x\sim\pi_e}[D(x)] - \mathbb{E}_{y\sim\pi_g}[D(y)], \tag{21}$$

where the supremum is over all the 1-Lipschitz functions $D(\cdot)$ (*i.e.,* the gradient of $D(\cdot)$ is not bigger than 1). Moreover, we define $D_\theta(\cdot)$ and $G_\phi(\cdot)$ as a discriminator and a generator which are respectively represented by neural networks with parameter θ and ϕ. To learn the generator's distribution π_g over real data $x \sim \pi_e$, we define a generated policy on input noise variables $z \sim \pi_z$, then represent a mapping to generated data space as $G_\phi(z)$. Moreover, $D_\theta(G_\phi(z))$ outputs a scalar within $[0, 1]$ which represents the probability that x comes

from the real data rather than π_g. Then, we consider solving the problem

$$W(\pi_e, \pi_g) = \max_{\theta:\|D_\theta\|_L \leq 1} \mathbb{E}_{x\sim\pi_e}[D_\theta(x)] - \mathbb{E}_{z\sim\pi_z}[D_\theta(G_\phi(z))]. \tag{22}$$

In detail, parameters θ and ϕ of the discriminator and generator are respectively updated by implementing the m-batchsize gradient descend, and we have

$$\nabla_\theta W(\pi_e, \pi_g) = \nabla_\theta[\frac{1}{m}\sum_{i=1}^{m}[D_\theta(x^{(i)}) - D_\theta(G_\phi(z^{(i)}))]]. \tag{23}$$

$$\nabla_\phi W(\pi_e, \pi_g) = -\nabla_\phi[\frac{1}{m}\sum_{i=1}^{m} D_\theta(G_\phi(z^{(i)}))]. \tag{24}$$

Obviously, (23) and (24) respectively update the parameters of θ and ϕ towards opposite directions. In particular, the objective of discriminator D is to discriminate the generated dataset and the real one to the greatest extent, while the generator G tends to minimize the possibility of being discriminated by the discriminator D.

As shown in Fig. 2, an experience replay buffer is used to store the training dataset. By mixing the real dataset and the generated adversarial dataset into the replay buffer, the correlation between the datasets can be decreased, which can help Double DQN improve the generalization and robustness. We summarize the process of W-GANs in Algorithm 1.

Algorithm 1 W-GANs for data sample expansion. All experiments in the paper use the default values $\alpha_g = 0.0005$, $\alpha_d = 0.0005$, $c = 0.01$, $m = 64$.

Input: α_g, the learning rate of generator; α_d, the learning rate of discriminator. c, the clipping parameter; m, the batch size; N_d, the number of training steps of the discriminator; N_g, the number of training steps until convergence of the generator; θ, initial parameters of the discriminator. ϕ, initial parameters of the generator.

Output: Adversarial samples generated by the generator G

while $n_g \leq N_g$ **do**
  for $n_d = 1, \cdots, N_d$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \pi_e$ a batch from the expert dataset.
    Sample $\{z^{(i)}\}_{i=1}^m \sim \pi_z$ a batch from the prior noise samples.
    $\dot{\theta} \leftarrow \nabla_\theta[\frac{1}{m}\sum_{i=1}^m[D_\theta(x^{(i)}) - D_\theta(G_\phi(z^{(i)}))]]$
    $\theta \leftarrow \theta + \alpha_d \cdot \mathrm{RMSProp}(\theta, \dot{\theta})$
    $\theta \leftarrow \mathrm{clip}(\theta, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim \pi_z$ a batch of the prior noise samples.
  $\dot{\phi} \leftarrow -\nabla_\phi[\frac{1}{m}\sum_{i=1}^m D_\theta(G_\phi(z^{(i)}))]$
  $\phi \leftarrow \phi - \alpha_g \cdot \mathrm{RMSProp}(\phi, \dot{\phi})$
  $n_g = n_g + 1$
end while

B. Few-Shot RL Procedure by Double DQN

Recently, many algorithms have been proposed to combine reinforcement learning with deep learning (*i.e.,* neural network), such as Actor-Critic [38], Asynchronous Advantage Actor-Critic [39], Deep Q-Network [40], Deep Deterministic Policy Gradient [41], *etc*. We choose the Double DQN algorithm which is improved based on the Natural DQN or Conventional DQN to implement the discrete-decision making process. The Double DQN is proposed to overcome the disadvantages of Natural DQN which is known to overestimate action values under certain conditions [18]. Generally, Double DQN consists of two networks: evaluation network and target network, which are parameterized by vectors of η and η^- respectively, where the greedy policy is evaluated according to the evaluation network, and the value is estimated by the target network.

The RL procedure is employed in the distributed network with a γ-discounted finite horizon setting, and follows the deployment of POMDP in Section III. Let $\mathcal{J}$ denote the RL procedure's objective of SBS b, which is given by

$$\mathcal{J} = \mathbb{E}_{X_b\sim\mathcal{X}, A_b\sim\mathcal{A}}[f(X_{b,t}, A_{b,t})]. \tag{25}$$

Let $\Phi(X)$ denote the feature vector, and the state-action value Q defined in (9) be parameterized by a vector η with the same dimension. Assume that the RL procedure of SBS b starts at network state $X_{b,t}$, and (9) turns into a state-action value $Q(X, A)$. The evaluated state-action value Q^{eval} is approximated as $Q^{eval}(X_{b,t}, A_{b,t}) \approx Q(X_{b,t}, A_{b,t}; \eta)$, and the target state-action value Q^{tar} is then approximated as $Q^{tar}(X_{b,t}, A_{b,t}) \approx Q(X_{b,t}, A_{b,t}; \eta^-)$. In detail, we choose the non-linear based neural network to approximate the two networks in case of the non-linear characteristics of the state-value functions. Thus, we have $Q^{eval}(X_{b,t}, A_{b,t}; \eta) = \eta^T \cdot \Phi(\mathrm{x})$, and $Q^{tar}(X_{b,t}, A_{b,t}; \eta^-) = \eta^{-T} \cdot \Phi(\mathrm{x})$.

Generally, there are two methods used to compute temporal difference (TD): Specifically, the forward method combines the future steps for joint optimization. As we pursue a fast reinforcement learning iteration in this work, and the states in the future two or more time steps may be observed in tens of TTIs in this scenario, we implement an one-step backup (*i.e.,* TD(0)) for adapting the time-varying network environment. Therefore, in order to update the state-action value Q, we let the TD-error be calculated as below

$$\delta_{i,(t)} = Q^{eval}(X_{b,t}, A_{b,t}; \eta) - Q^{tar}(X_{b,t}, A_{b,t}; \eta^-), \tag{26}$$

where the target network is updated by

$$Q^{tar}(X_{b,t}, A_{b,t}; \eta^-) = R_{b,t+1} + \gamma \cdot Q^{tar}(X_{b,t+1}, \arg\max_{A_{b,t+1}} Q^{eval}(X_{b,t+1}, A_{b,t+1}; \eta); \eta^-), \tag{27}$$

in which the parameters of the target network stays unchanged from DQN, and remains a periodic copy of the online network.

Then, let the loss function for SBS b at time t be denoted by $\mathcal{L}_{b,t}(\eta)$. During the learning procedure, we aim to minimize the loss function below

$$\arg\min_{\eta} \mathcal{L}_{b,t}(\eta) = \mathbb{E}_{X_b\sim\mathcal{X}, A_b\sim\mathcal{A}}|\delta_{i,(t)}|^2, \tag{28}$$

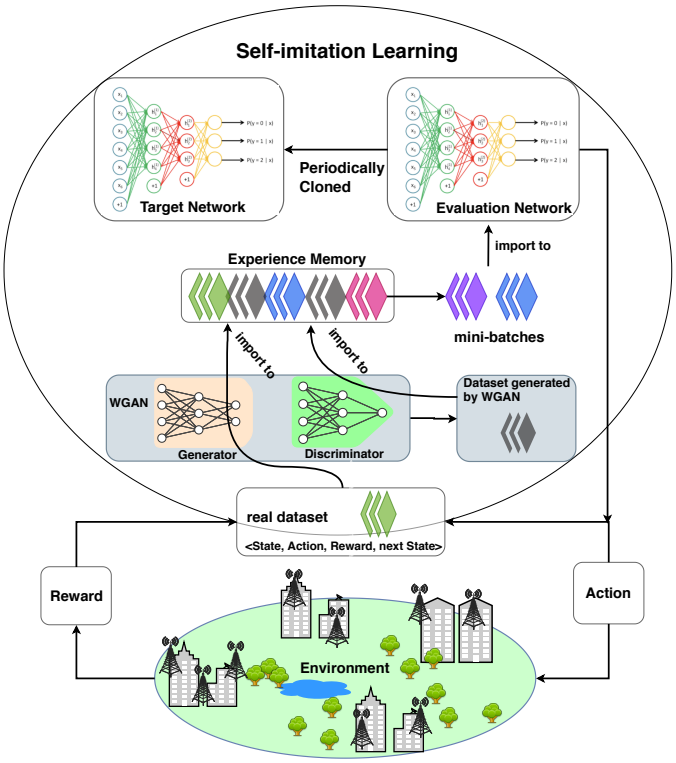


Fig. 2. Self-imitation Learning Framework

Since the action space is discrete, we use Softmax layer for selecting an action. The Softmax function is used as a continuous, differentiable approximation to $\arg\max$, and which is defined as $\text{Softmax}(x_i) = e^{x_i}/\sum_j e^{x_j}$. Generally, the discrete action can be sampled from a multinomial distribution with probabilities given by the output of a *Softmax* function. Nevertheless, the resulted sampling process is not differentiable. Thus we obtain a differentiable approximation by sampling from the Gumbel-Softmax distribution [42] which has been previously used to train variational auto-encoders with discrete latent variables [43].

Since there are K RBs, the Softmax layer is designed to have K outputs variables. Let the output probabilities of $A_{b,t} = k$ be denoted as π_k. Then,

$$A_{b,t} = \arg\max_k(\log(\pi_k) + \mathcal{G}_k(u)), \tag{29}$$

where the variables in vector $\mathcal{G} = [\mathcal{G}_1(u), \cdots, \mathcal{G}_k(u)]^T$ are *i.i.d* samples drawn from Gumbel distribution. In detail, the Gumbel distribution can be sampled using inverse transform sampling by drawing $u \sim U(0,1)^K$ (let $U(0,1)^K$ be the K-dimensional uniform distribution on the interval $[0,1]$) and computing $\mathcal{G}(u) = -\log(-\log(u))$.

Moreover, we introduce ς as a controllable inverse temperature hyper-parameter in the Softmax function. Then (29) is rewritten as

$$A_{b,t} = \text{softmax}(\frac{\log(\pi_\text{k}) + \mathcal{G}_\text{k}(\text{u})}{\varsigma}), k = 1, \cdots, K. \tag{30}$$

When $\varsigma \to 0$, the Softmax layer acts like *argmax* (*Softmax*$\sim$*argmax*) resulting in low bias while the variance of the gradient of the Softmax increases. On the other hand, when ς is set a little larger, the *Softmax* creates a smoothing effect while the bias turns to be high (*Softmax*$\neq$ *argmax*). Therefore, at the beginning of the training, we set ς to a large value, so that the gradient flow is smoother. Later we let ς approach 0, so that the vector obtained by Softmax is closer to the result of *argmax*.

The SIL framework consists of WGANs and Double DQN where the algorithm of WGANs is trained in the learning process of the Double DQN. We elaborate the learning process of the SIL in Algorithm 2 for ease of understanding.

C. Computational Complexity

For family SBSs (or Femto) equipped with low power and low CPU frequency, an algorithm with low computational complexity is essential. We analyze the computational complexity of the proposed SIL algorithm. Specifically, according to the learning process summarized in Algorithm 2, the computational complexity of SIL denoted by $\mathcal{O}_{SIL}$ is evaluated by jointly considering the computational complexity of W-GANs and Double DQN, which is given by

$$\mathcal{O}_{SIL} = \mathcal{O}_{WGANs} + \mathcal{O}_{DDQN} \tag{31}$$

According to Algorithm 1, we first give the computational complexity of the WGANs algorithm

$$\mathcal{O}_{WGANs} = \mathcal{O}(N_g \cdot N_d). \tag{32}$$

Therefore, for one operation epoch, the computational complexity of the SIL is derived as

$$\mathcal{O}_{SIL} = \mathcal{O}(N_o \cdot (N_g \cdot N_d + N_c + N_m)). \tag{33}$$

Algorithm 2 Self-imitation learning algorithm

Input: α_D, the learning rate of the Double DQN; N_m, the number of time steps to save policy trajectories into the experience replay buffer; N_o, the overall operation epochs; N_c, the number of training steps to the convergence of Double DQN. N_u, the number of training steps to replace η^- by η.

Output: Optimal resource allocation strategies set $\mathcal{A}^*$

Initialize evaluated action-value function Q^{eval} with random weights η.
Initialize target action-value function Q^{tar} with weights $\eta^- = \eta$.
Initialize sequence $x = \{X_t\}$ and feature vector $\Phi(X_t)$.
for $n_o = 1, \cdots, N_o$ **do**
  for $n_m = 1, \cdots, N_m$ **do**
    Execute action A_t under the observable state X_t. Then feedback the instantaneous reward R_t, state X_{t+1} and the feature vector $\Phi(X_{t+1})$.
    Store the transition sequence $(\Phi(X_t), A_t, R_t, \Phi(X_{t+1}))$ in the replay buffer.
  end for
  Train the WGANs, and import the generated dataset to the replay buffer after the convergence of WGANs.
  for $n_c = 1, \cdots, N_c$ **do**
    Sample random mini-batches of transitions $(\Phi(X_j), A_j, R_j, \Phi(X_{j+1}))$ from the experience replay buffer.
    Set

$$Q_j = \begin{cases} R_j, \text{if episode terminates} \\ R_j + \gamma \max_{a'} \hat{Q}(\Phi(X_{j+1}), A'; \eta^-), \text{otherwise} \end{cases}$$

    Perform a gradient descent step on $\mathcal{L}_{b,t}(\eta)$ with respect to the evaluation network parameter η.

$$\eta \leftarrow \eta - \alpha_D \cdot \nabla_\eta \mathcal{L}_{b,t}(\eta)$$

    if $\mathrm{mod}(N_c, N_u) \neq 0$ **then**
      Reset $Q^{tar} = Q^{eval}$ and let $\eta^- = \eta$.
    end if
  end for
end for

From (33), we can see that the SIL provides a polynomial time solution for solving the problem (16). Moreover, from the simulation results, we can see that the WGANs and Double DQN can converge after thousands of iterations in the current state. Therefore, SIL is able to provide an efficient solution with low computational complexity for making TTI level decisions.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed SIL by extensive simulations. We use the Multi-Wall-and-Floor (MWF) model [44] as the propagation and penetration loss model between SBS and UE in our indoor scenario. MWF takes into account the decreasing penetration loss of walls and floors of the same category as the number of traversed walls/floors increase, which is given by $PL(d)[dB] = L_0[dB] + 20\,log_{10}(d) + n_w L_w$, where L_0 is the reference loss [dB] taken at one meter of distance between the transmitter and the receiver, d is the distance between SBS and UE in meters, $L_w = 6dB$ is the penetration loss of the concrete wall, n_w is the number of walls. Other parameters used are listed in TABLE II.

TABLE II
Simulation parameters

Parameter Description	Value
System Bandwidth	20 MHz
Number of Smart BSs	20
Number of Normal BSs	30
Number of UEs under a BS	10
RB Bandwidth	180 KHz
Noise power spectral density	-174 dBm/Hz
Maximum SBS Transmit Power	23 dBm
Number of RBs	100
Resource allocation interval	1 TTI (1ms)
Reward Discount γ	0.99
Replay Buffer	10000

A. Comparison References in the Simulation

We use the following algorithms as the comparison references in our performance evaluation:

1) *Natural DQN based ICIC*: Since the action space of the POMDP is discrete, we use Double DQN for implementing the discrete decision-making process of the ICIC. Particularly, there is no target network in the Natural DQN, and the evaluated state-action value Q is updated without latency.
2) *Double DQN based ICIC*: Double DQN consists of two networks: evaluation network and target network, where the greedy policy is evaluated according to the evaluation network, and the evaluated Q value is estimated by the target network. By this way, it can overcome the overestimation problem in the discrete decision-making process.
3) *MaxSINR based ICIC*: MaxSINR chooses actions (i.e., subcarrier) with the largest SINR, which is a greedy policy without focusing on long-term network performance.
4) *SIL based ICIC*: SIL is the proposed self-imitation learning algorithm in this work.

B. Numerical Results and Discussion

We first examine the convergence and the performance of the generator and discriminator of WGANs. Fig. 3 shows the Wasserstein estimation varying with the training epochs. From the simulation results, we can see that the discriminator and generator are trained in the opposite direction. This is because the discriminator and the generator are respectively updated according to (23) and (24) which have the opposite objectives. We can see that both the two training curves converge after about 2000 epochs. The Wasserstein estimation of the generator converges to 1, which means the adversarial

dataset generated by the generator G cannot be discriminated by the discriminator D. Moreover, the Wasserstein estimation of the discriminator finally converges to 0, which means D is impossible to distinguish between the generated dataset and the real dataset correctly.

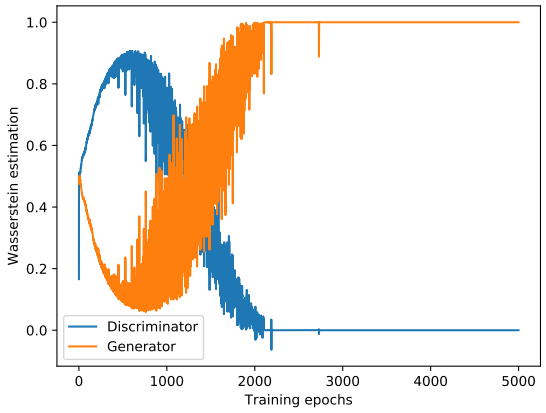


Fig. 3. Training process of WGANs

Next, we compare the ability of the Natural DQN and the Double DQN in solving the overestimation problem in the discrete decision-making process. Fig. 4 shows that the evaluated state-action value Q changes with the increase of the training steps. From the simulation results, we can see that with the increase of the training steps, the changes of the two curves gradually stabilized after about 5000 training steps. This illustrates that the convergence of the two algorithms is quite similar. Since the evaluated Q value converges around 75, the Natural DQN obviously overestimate the evaluated Q value before about 4000 training steps where the evaluated Q value of the Natural DQN is always larger than 75. The overestimation problem of the Natural DQN will result in a significant error when estimating the evaluated Q value, which makes the decision-making of the RL procedure inaccurate. Moreover, we can see that the convergence area of the Double DQN is slightly larger than that of the Natural DQN, which means the Double DQN can finally converge to better strategies. Therefore, we choose Double DQN in the proposed SIL framework for performing the resource allocation of ICIC.

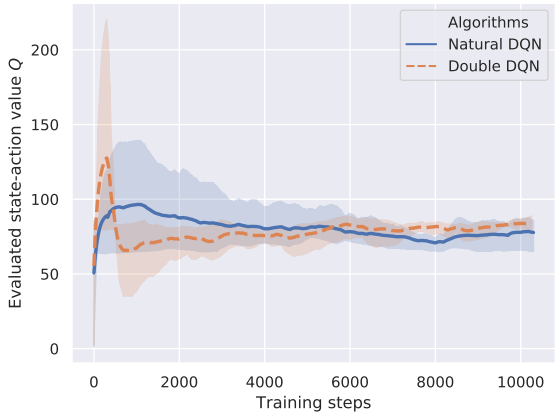


Fig. 4. The updating process of the evaluated state-action value Q

Then, we illustrate the training process of SIL over time steps. Fig. 5 shows the evaluated Q value which changes with the training steps. As shown in Fig. 5, the training process has two continuous stages: buffer stage and training stage. In particular, in the buffer stage, the training dataset shaped with $\langle state, action, action', state' \rangle$ are collected during the operation period until the buffer size is full. Then, in the training stage, the training dataset from the buffer is used for implementing the batch-size policy gradient. After the convergence of the training process, the derived strategies are used for performing resource scheduling in the SIL.

In order to cater for the plug-and-play operation mode of SBSs, we adopt a nested training scheme to reduce the performance degradation occurring in the slow-start period and thus to make the training process of SIL smoothy. Fig. 6 shows the execution process of the proposed SIL algorithms, which consists of two processes: a slow-start process and normal execution process. Specifically, the slow-start process happens at the boot-up stage of an SBS, where the SIL is unavailable to be used since the replay buffer is in loading process and the algorithm has not yet converged. After the completion of the first slow-start process, the normal execution process begins, where the next slow-start process and the normal execution process work in a parallel way until the slow-start process converges. By this way, SIL's execution process is relatively smoothy and the training time in the replay buffer can be significantly reduced.

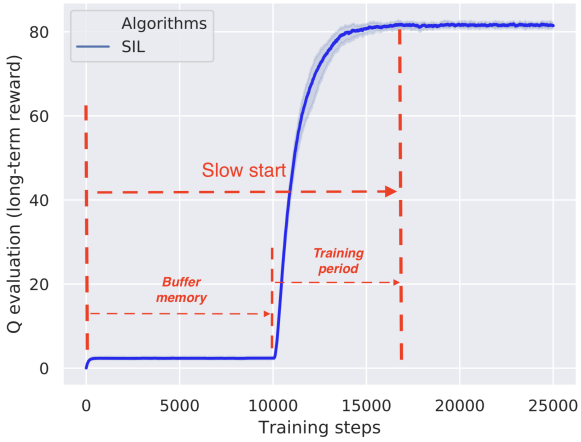


Fig. 5. Training process: the slow start and convergence of SIL

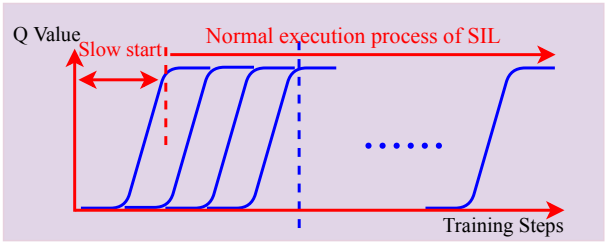


Fig. 6. Nested training in SIL

Next, we examine the performance of SIL framework on the average throughput with and without the maxSINR initializa-

tion in the slow-start area. SIL with maxSINR initialization means the index in the Softmax output layer is initialized according to the value of SINR. Fig. 7 shows the average throughput over time steps in the slow-start phase of the SIL. From the simulation result, we can see that in the slow-start phase, the performance on the average throughput of SIL with maxSINR initialization is better than that of SIL without maxSINR initialization. The average throughput is significantly improved about 38.5%. Therefore, the design of SIL with maxSINR initialization can effectively improve the performance of SIL in the slow-start phase.

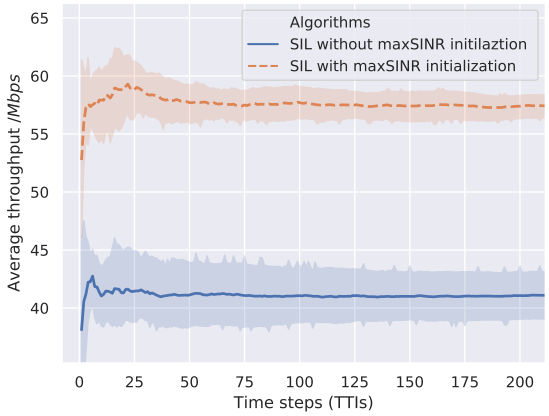
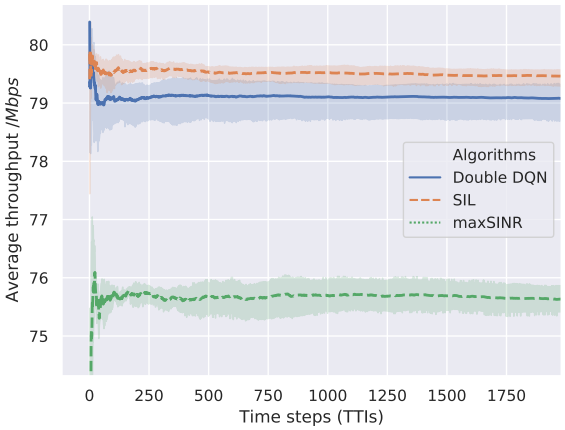


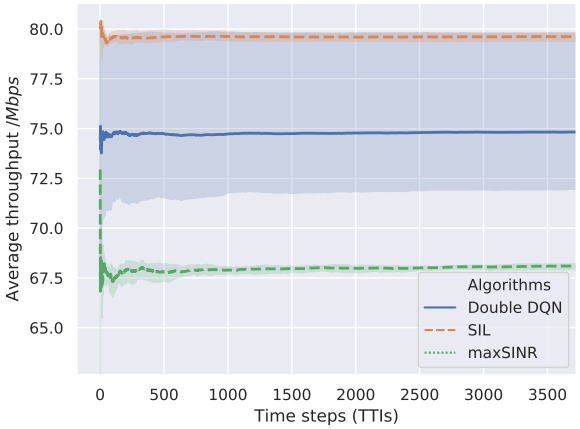
Fig. 7. Performance comparison of SIL with and without maxSINR initialization in the slow-start phase

Next, we extend the simulation to different network settings, and examine the average throughput of the three algorithms over time steps (TTIs). Specifically, we consider the coexistence of smart SBSs and legacy SBSs running the maxSINR ICIC in the HetNet, and fix the number of smart SBSs to 1 in this simulation. Two simulation settings are used for comparison: 1) Simulation implemented in Fig. 9(a) considers 10 legacy SBSs in the environment, and 2) Simulation implemented in Fig. 9(b) considers 50 legacy SBSs in the environment. From Fig. 9(a) and Fig. 9(b), we can see that the average throughput of SIL is higher than that of the Double DQN and the maxSINR. In summarize, in Fig. 9(a), the average throughput of SIL is improved about 0.6% and 4.8% when compared with Double DQN and maxSINR. In Fig. 9(b), the improvement is about 6% and 18.5%. In the simulation settings of Fig. 9(a), since the number of subcarriers is adequate for a small number of SBSs, Double DQN and maxSINR can easily find good strategies with great probability. Therefore, as the environment becomes more complex, the improvement of the performance of SIL on the average throughput becomes more significant when compared with that of Double DQN and maxSINR.

In the last experiment, we compare the overall throughput of the Double DQN, SIL and maxSINR under different number of SBSs with fixed 50 environmental legacy SBSs. From the simulation result of Fig. 9, we can see that the overall throughput increases with the number of SBSs. In more detail, when the number of the SBSs is less than 2, the difference between the three algorithms is not obvious, which is because the spectrum resources are adequate for a small number of



(a) Simulation setting 1



(b) Simulation setting 2

Fig. 8. Average throughput with different environment UEs in a smart BS

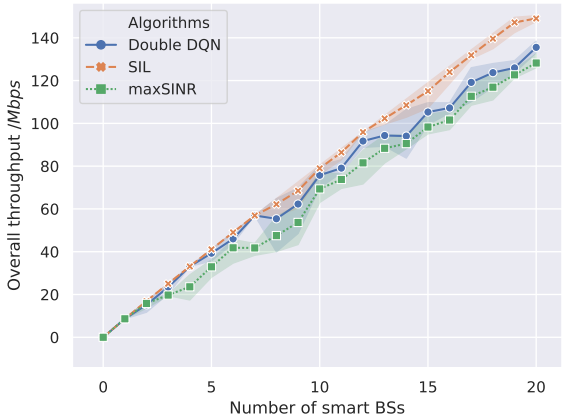


Fig. 9. Overall throughput of all smart BSs

SBSs. When the number of SBSs increases to larger than 2, the performance improvement of the SIL becomes obvious when compared with that of other algorithms. Specifically, the average improvement of SIL on the overall throughput is about 9.9% and 19.8% when compared with that of Double DQN and maxSINR. Therefore, we can see that the SIL can significantly improve the performance compared with the reference algorithms especially in more complex network environments.

VII. Conclusion

In this paper, we have proposed a self-imitating learning (SIL) for solving the ICIC problem in decentralized HetNets. The decision-making processes of the decentralized SBSs are modeled as non-cooperative POMDP games, and the objective of the decision-making process is formulated as a distributed IRL problem aiming at improving the long-term performance on system throughput while guaranteeing the QoS requirements. The main idea of SIL which consists of W-GANs and Double DQN is used for performing behavior imitation and few-shot learning with the aim of optimizing the IRL problem from both the Policy and Value. In more detail, Double DQN can significantly eliminate the overestimation and perform the decentralized resource scheduling in this work, and W-GANs in SIL are used to imitate the expert strategies and generate adversarial training samples, and thus help Double DQN make decisions more robust. Significant performance improvements in terms of the average throughput and overall throughout are achieved by SIL when compared with other benchmark algorithms.

References

[1] C. V. N. Index, "global mobile data traffic forecast update, 2017–2022," *Cisco White paper*, 2019.

[2] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5g: challenges, methodologies, and directions," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 78–85, 2016.

[3] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2522–2545, 2016.

[4] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5g ultra-dense cellular networks," *arXiv preprint arXiv:1512.03143*, 2015.

[5] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 56–62, 2019.

[6] M. S. Haroon, Z. H. Abbas, F. Muhammad, and G. Abbas, "Coverage analysis of cell-edge users in heterogeneous wireless networks using stienen's model and rfa scheme," *International Journal of Communication Systems*, vol. 33, no. 10, p. e4147, 2020, e4147 dac.4147. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4147

[7] F. Ahmed, A. A. Dowhuszko, and O. Tirkkonen, "Self-organizing algorithms for interference coordination in small cell networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8333–8346, 2017.

[8] Y. Li, M. Sheng, Y. Sun, and Y. Shi, "Joint optimization of bs operation, user association, subcarrier assignment, and power allocation for energy-efficient hetnets," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3339–3353, 2016.

[9] C. Pan, M. Elkashlan, J. Wang, J. Yuan, and L. Hanzo, "User-centric c-ran architecture for ultra-dense 5g networks: Challenges and methodologies," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 14–20, 2018.

[10] V. Sciancalepore, I. Filippini, V. Mancuso, A. Capone, and A. Banchs, "A multi-traffic inter-cell interference coordination scheme in dense cellular networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 5, pp. 2361–2375, 2018.

[11] C. Yang, J. Xiao, J. Li, X. Shao, A. Anpalagan, Q. Ni, and M. Guizani, "Disco: Interference-aware distributed cooperation with incentive mechanism for 5g heterogeneous ultra-dense networks," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 198–204, 2018.

[12] J. Yoon and G. Hwang, "Distance-based inter-cell interference coordination in small cell networks: stochastic geometry modeling and analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4089–4103, 2018.

[13] Z. H. Abbas, M. S. Haroon, G. Abbas, and F. Muhammad, "Sir analysis for non-uniform hetnets with joint decoupled association and interference management," *Computer Communications*, vol. 155, pp. 48–57, 2020.

[14] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2014.

[15] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Pérez, "Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 3044–3058, 2017.

[16] J. Zhao, T. Q. Quek, and Z. Lei, "Coordinated multipoint transmission with limited backhaul data transfer," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2762–2775, 2013.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[18] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[20] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI conference on artificial intelligence*, Conference Proceedings.

[21] C. Xiong, G. Y. Li, S. Zhang, Y. Chen, and S. Xu, "Energy-and spectral-efficiency tradeoff in downlink ofdma networks," *IEEE transactions on wireless communications*, vol. 10, no. 11, pp. 3874–3886, 2011.

[22] J. Tang, D. K. So, E. Alsusa, and K. A. Hamdi, "Resource efficiency: A new paradigm on energy efficiency and spectral efficiency tradeoff," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4656–4669, 2014.

[23] X. Ge, X. Li, H. Jin, J. Cheng, and V. C. Leung, "Joint user association and user scheduling for load balancing in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3211–3225, 2018.

[24] A. Ullah, Z. Haq Abbas, G. Abbas, F. Muhammad, and L. Jiao, "Performance analysis of user-centric sbs deployment with load balancing in heterogeneous cellular networks: A thomas cluster process approach," *Computer Networks*, vol. 170, p. 107120, 2020.

[25] R. D. Armstrong, D. S. Kung, P. Sinha, and A. A. Zoltners, "A computational study of a multiple-choice knapsack algorithm," *ACM Transactions on Mathematical Software (TOMS)*, vol. 9, no. 2, pp. 184–198, 1983.

[26] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5g radio access network slicing," *IEEE Transactions on Vehicular Technology*, 2019.

[27] R. D. Armstrong, D. S. Kung, P. Sinha, and A. A. Zoltners, "A computational study of a multiple-choice knapsack algorithm," *ACM Transactions on Mathematical Software (TOMS)*, vol. 9, no. 2, pp. 184–198, 1983.

[28] M. Yan, G. Feng, J. Zhou, and S. Qin, "Smart multi-rat access based on multiagent reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4539–4551, 2018.

[29] R. Shi, J. Zhang, W. Chu, Q. Bao, X. Jin, C. Gong, Q. Zhu, C. Yu, and S. Rosenberg, "Mdp and machine learning-based cost-optimization of dynamic resource allocation for network function virtualization," in *2015 IEEE International Conference on Services Computing*. IEEE, 2015, pp. 65–73.

[30] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric iot," *IEEE Transactions on Emerging Topics in Computing*, 2018.

[31] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[32] F. Liu, E. Bala, E. Erkip, M. C. Beluri, and R. Yang, "Small-cell traffic balancing over licensed and unlicensed bands," *IEEE transactions on vehicular technology*, vol. 64, no. 12, pp. 5850–5865, 2015.

[33] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on networking*, no. 5, pp. 556–567, 2000.

[34] E. Maskin, "Nash equilibrium and welfare optimality," *The Review of Economic Studies*, vol. 66, no. 1, pp. 23–38, 1999.

[35] A. M. Fink *et al.*, "Equilibrium in a stochastic n-person game," *Journal of science of the hiroshima university, series ai (mathematics)*, vol. 28, no. 1, pp. 89–93, 1964.

[36] A. Y. Ng, S. J. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[37] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations (Grundlehren der Mathematischen Wissenschaften)*. Springer Berlin, 1971, vol. 170.

[38] V. R. Konda and N. Tsitsiklis, "Actor-critic algorithms," *Siam Journal on Control & Optimization*, vol. 42, no. 4, pp. 1143–1166, 2003.

[39] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a gpu," *arXiv preprint arXiv:1611.06256*, 2016.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[41] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *ICML*, 2014.

[42] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[43] M. J. Kusner and J. M. Hernández-Lobato, "Gans for sequences of discrete elements with the gumbel-softmax distribution," *arXiv preprint arXiv:1611.04051*, 2016.

[44] M. Lott and I. Forkel, "A multi-wall-and-floor model for indoor radio propagation," in *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No. 01CH37202)*, vol. 1. IEEE, 2001, pp. 464–468.

Mu Yan received his B.Eng. degree in Electronic Engineering from the Beijing Jiaotong University, in 2014, and the Ph.D. degree from the National Key Lab of Science and Technology on Communications, University of Electronic Science and Technology of China (UESTC) in 2020. From 2018 to 2019, he was a Visiting Student with the Social Robotics Laboratory of Interactive Digital Media Institute, National University of Singapore, Singapore. He is currently an assistant research fellow in the Northern Institute of Electronic Equipment of China. Dr. Yan has extensive research experience and has published widely in wireless networking research. His research interests include intelligent wireless networking, network slicing, and resource management in mobile networks, etc.

Jian Yang received the B.S. and M.S. degrees in Communication and Information Systems from the Information Engineering University, Zhengzhou, China, in 2003 and 2006, respectively. After his graduation, he joined the Northern Institute of Electronic Equipment of China, where he is currently a Research Associate. He is mainly interested in communication signal processing.

Keyu Chen is going to receive his Bachelor degree of Engineering in School of Electronic and Information Engineering, Beijing Jiaotong University, in 2021, and pursue his Master's degree in School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research interests include array signal processing, cooperative working for UAVs, partical swarm optimization(PSO), etc.

Yao Sun received the B.S. degree in Mathematical Sciences, and the Ph.D. degree in Communication and Information System from University of Electronic Science and Technology of China (UESTC), in 2014 and 2019, respectively. He has published widely in wireless networking research area, and received the IEEE ComSoc TAOS Best Paper Award in 2019. His research interests include intelligent access control, handoff and resource management in mobile networks.

Gang Feng (M'01, SM'06) received his BEng and MEng degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong in 1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University in December 2000 as an assistant professor and became an associate professor in October 2005. At present he is a professor with the National Laboratory of Communications, UESTC. Dr. Feng has extensive research experience and has published widely in wireless networking research. A number of his papers have been highly cited. He has received the IEEE ComSoc TAOS Best Paper Award and ICC best paper award in 2019. His research interests include next generation mobile networks, mobile cloud computing, AI-enabled wireless networking, etc. Dr. Feng is a senor member of IEEE.