Deep Reinforcement Learning-Based Mobility-Aware Robust Proactive Resource Allocation in Heterogeneous Networks

Jing Li, Xing Zhang, Jiaxin Zhang, Jie Wu, Qi Sun, and Yuxuan Xie

***Abstract*—Proactive resource allocation (PRA) is an essential technology boosting intelligent communication, as it can make full use of prediction and significantly improve network performance. However, most promising gains base on perfect prediction which is unrealistic. How to make PRA robust against prediction uncertainty and maximize benefits brought by prediction becomes an important issue. In this paper, we tackle this problem and propose a mobility-aware robust PRA approach (MRPRA) in heterogeneous networks. MRPRA pre-allocates resources in both time and frequency domains among mobile users with users' trajectories predicted by hidden Markov model. The objective is to minimize service delay under constraints of different levels of quality-of-service (QoS) requirement and mobility intensity. MRPRA is robust against prediction uncertainty by exploiting probabilistic constraint programming to model QoS requirements in a probabilistic sense. To this end, the probabilistic distribution of achievable rate is derived. To flexibly coordinate resource allocation among multiple mobile users over time horizon, a deep reinforcement learning based multi-actor deep deterministic policy gradient algorithm is designed. It learns robust PRA policies by distributed acting and centralized criticizing. Extensive numerical simulations are performed to analyze performances of the proposed approach.**

***Index Terms*—Heterogeneous networks, proactive resource allocation, mobility prediction, deep reinforcement learning, robustness.**

I. Introduction

BIG DATA prediction makes the traditional heterogeneous networks (HetNets) learning and knowledgeable. It's an efficient way towards network intellectualization which is a dominant trend at present. 3GPP has introduced module of network data analytics into 5G systems to explore implicit intelligence from network data and guide the network towards efficient operation [1]. Promising technologies, e.g., mobile edge computing [2], caching [3], have incorporated big data prediction into performance enhancing as well.

Proactive resource allocation (PRA) is also an efficient approach boosting intelligent communication, as it can make full use of prediction and hence significantly improve network performance in terms of throughput, energy efficiency, quality-of-service (QoS) etc. PRA means to utilize some kinds of predicted information to make resource allocation planning beforehand for non-real-time (NRT) service [4]. This makes resource allocation process more flexible in large time scale. For example, if information of future wireless channel conditions and users' mobility is known, power and bandwidth allocation can be pre-designed to transmit more data when channel condition is good and available bandwidth is sufficient. This way helps to save energy consumption [5]. On the other hand, we can plan to first schedule those who are to leave the network's coverage to adapt to various levels of delay requirements in long term [6]. However, in traditional reactive schemes like fair scheduling (FS) in which users accessed to the same base station (BS) are scheduled with equal frequency bandwidth, the network reacts to arriving requests in a rigid way and hence lacks these functionalities. Thus, how to efficiently exploit predicted information for PRA optimization should be given comprehensive and deep exploration.

Besides, the promising gains mentioned above mostly base on perfect prediction. However, there always exist random prediction errors which bring randomness to the predicted information. Therefore, we say prediction is uncertain. Our previous work [6] has demonstrated that network performance is largely degraded by prediction uncertainty. The imperfectly predicted information will mislead PRA. It costs extra resources to complete service for the under-served users, which causes large service delay and low throughput. How to make PRA robust against prediction uncertainty to maximize benefits of prediction has not been thoroughly settled. And it poses challenges on modeling prediction uncertainty [7].

Inspired by the fact that human mobility and channel conditions are proven to be predictable [8], [9], this work exploits these two kinds of predicted information for PRA optimization. We also focus on effective processing prediction uncertainty to make PRA robust. A mobility-aware robust PRA (MRPRA) approach for NRT service is proposed. MRPRA

Manuscript received March 30, 2019; revised July 15, 2019 and October 11, 2019; accepted November 6, 2019. This work is supported by the National Science Foundation of China (NSFC) under grant 61771065, 61631005, 61901048, by Beijing Municipal Science and Technology Commission Research under Project Z181100003218015, and by Fundamental Research Funds for the Central Universities under Project 500418765. The associate editor coordinating the review of this article and approving it for publication was J. Hoydis. *(Corresponding author: Xing Zhang.)*

J. Li, X. Zhang, and J. Zhang are with the Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Communications, Beijing 100876, China (e-mail: ljing@bupt.edu.cn; hszhang@bupt.edu.cn; jxz@bupt.edu.cn).

J. Wu, Q. Sun, and Y. Xie are with the Green Communication Research Center, China Mobile Research Institute, Beijing 100053, China (e-mail: wujieyjy@chinamobile.com; sunqiyjy@chinamobile.com; xieyuxuan@chinamobile.com).

Digital Object Identifier 10.1109/TCCN.2019.2954396

aims to minimize service delay under constraints of different QoS requirements and mobility intensities by optimally coordinating allocation of time slots and frequency bandwidth.

Solving the robust PRA optimization problem is challenging. First, the problem is mixed integer and non-convex. Second, the problem complexity sharply increases with the size of prediction window. Third, robust PRA is performed across multiple BSs and mobile users over time horizon under coexistence of different levels of QoS requirement and mobility intensity, which reflects complexity of the environment. Deep reinforcement learning (DRL) is an efficient tool to overcome those challenges [10]. The agent is trained to make decisions sequentially by learning from the environment to maximize its reward in long term. Taking advantage of this feature, we can decompose the original problem space into much smaller subspaces and train the agent to make optimal decisions in these subspaces sequentially. The agent gets an approximately optimal solution in the original problem space by maximizing the long term reward.

The major contributions of this work include:

- By assuming that perfectly predicted users' mobility and channel gains are known, we model the PRA optimization problem to provide a performance upper bound. A weight is designed for each user to adapt to their QoS requirements and mobility intensities. As the running time of directly solving the optimization problem largely grows with the size of prediction window, we decompose the problem in a prediction window into sub-optimization problem in each frame and iteratively update solutions until convergence.
- Each user's mobility trace is predicted by hidden Markov model (HMM). In order to maximize benefits of prediction, probabilistic constrained programming (PCP) is utilized to make PRA robust against prediction uncertainty by modeling QoS requirement constraints in a probabilistic sense. To this end, prediction uncertainty of users' mobility traces and channel gains is translated into rate uncertainty of which probabilistic density function (PDF) is derived. Since rate distribution is utilized, it doesn't need to predict exact realizations of channel gains.
- Robust PRA optimization is further modeled as a Markov decision process (MDP). We solve the problem for each time slot sequentially instead of simultaneously determining all variables in the whole prediction window. In this way, the complexity is significantly reduced. An actor–critic based DRL algorithm — deep deterministic policy gradient (DDPG) is introduced. In order to flexibly coordinate resource allocation among multiple users over time horizon, we extend DDPG to multi-actor DDPG to make robust PRA decision in a way of distributed acting and centralized criticizing. A reward function that prompts actors to complete their transmissions is designed to help the critic evaluate each actor's policy.

The rest of this paper is organized as follows. Section II reviews the related work. Section III gives the system model. Section IV models PRA with and without perfect prediction, and elaborates how to use PCP to handle prediction uncertainty. In Section V, robust PRA optimization is modeled as MDP and solved by our designed multi-actor DDPG algorithm. Section VI explores benefits of utilizing predicted information and evaluates the performance of the proposed approach by simulations. Comprehensive conclusion is given in Section VII.

II. Related Work

A. Resource Allocation Planning With Prediction

On condition that the network has perfectly predicted the arrival time and contents of users' requests ahead of time, works in [11], [12] proposed to activate the BS to pre-download files from the core network before users' requests actually arriving. Performance gains of the proactive policy come from extending the transmission deadline and hence shrinking the queue length.

Works in [4], [6], [13]–[15] pre-allocated resources in a prediction window. They assumed that perfectly predicted information on user mobility, channel conditions and traffic demands was available at the beginning of the prediction window. Work in [4] studied how to translate the predicted information to available knowledge for planning power and time slots allocation, BS sleeping. Work in [13] proposed two approaches for tradeoff between power consumption and service delay. The approach with future information can optimize resource allocation for multiple frames but the one without can work for only one frame. This indicates that proactive schemes help optimize resource management in large time scale. Work in [14] minimized the maximal service delay for time slots allocation planning with a heuristic algorithm. Work in [15] minimized energy consumption by optimizing power allocation according to predicted channel conditions. Our previous work [6] studied how and when to utilize perfect prediction for PRA with convex optimization. This work fully developed our early ideas in [6]. The differences between them include, a) the previous work solved PRA optimization problem directly, while this work decomposed the problem space and utilized an iterative solver for computational complexity reduction, b) the previous work assumed perfect prediction, while this work predicted user mobility and further utilized PCP to make PRA robust against prediction uncertainty, c) this work additionally designed a DRL algorithm to tackle challenges in solving robust PRA optimization problem.

The above researches are based on perfect prediction. But prediction is always uncertain practically. One of our major contribution compared to the above works is that we dealt with prediction uncertainty. PCP is one of the main technologies in stochastic programming to tackle uncertainty and provide robust information. The predicted uncertain values are represented as stochastic variables [16]. And the constraints accommodating the predicted uncertain values are presented in a probabilistic form with a maximum violation probability. Works in [7] and [17] proposed to use PCP to model resource allocation in predictive video streaming. These works only considered rate uncertainty and assumed perfectly known mobility traces. However, the bias in mobility prediction brings wrong knowledge of user association and causes a waste of resources, which is ineligible. In contrast, our work incorporated mobility prediction uncertainty.

There are other PRA researches based on mobility prediction models. Work in [5] designed four deep neutral networks (DNN) to predict user mobility, thresholds of average channel gain and average residual frequency bandwidth to guide data transmission. Work in [18] proposed a resource reservation method by predicting users' next locations based on decision tree and Markov model. Work in [19] proposed a proactive BS sleep cycle scheduling scheme with help of the designed next location estimation algorithm. Performance of these works largely depend on prediction accuracy. In contrast, our work achieved robustness against inaccurate prediction. Moreover, we considered adaptiveness to mobility intensities.

B. User Mobility and Channel Gain Prediction

In order to get future knowledge of user mobility, effort in [20] proposed a mobility prediction framework based on hidden Markov model (HMM). The spatio-temporal predictor derived the future travel sequence given a future time sequence. The probabilistic distribution of users' future positions can be obtained with HMM as well. Work in [21] used recurrent neural network to predict the next visited cell. A long short-term memory based human path predictor was proposed in [22]. As our work focused on exploring the value of users' future moving traces, and we only needed coarse future positions together with their probabilistic distribution, we adopted the framework in [20] for mobility prediction. Work in [23] provided various ways to predict channel gains with the help of a coverage map which can be constructed with [9]. However, as we utilized PCP to model rate uncertainty in a probabilistic sense, there is no need to directly predict future channel conditions.

C. Reinforcement Learning for Resource Allocation

Uncertain dynamic wireless environment, demand of adaption to diverse users' behaviors have posed challenges on resource allocation. More and more studies utilized RL to tackle those challenges recently. An MDP based online learning method was proposed in [24] for MEC offloading. The state transition probability it used is often hard to obtain. Other works focused on model-free algorithms. Work in [25] proposed a user association approach with deep Q-network (DQN). However, handling continuous action space is beyond the capability of DQN. Work in [26] proposed a user association and power allocation scheme based on actor–critic learning framework. The linear feature-based function it used may not provide good estimation of the action-value function when the environment is complex. DDPG algorithm [27] combined DQN and the actor–critic framework to handle continuous state and action spaces. It utilizes DNN to approximate the action-value function and policy function, which has good adaptiveness to complex environment. In this work, we borrowed the idea from [29] to extend DDPG to multi-actor DDPG. Work in [29] proposed a novel multi-actor framework and made each actor execute a distinct task. While in our work, all actors cooperated to complete the same task.

TABLE I
SUMMARY OF MAIN NOTATIONS

Notation	Description
c_u^{min}	Minimum data rate requirement of user u
$\varphi_{t,u}$	The BS that user u associates with in time slot t
$\hat{\varphi}_{t,u}$	Predicted value of $\varphi_{t,u}$
$\gamma_{t,u}$	Spectrum efficiency of user u in time slot t
$R_{t,u}$	Total available frequency bandwidth for user u in time slot t with stochastic form $\tilde{R}_{t,u}$
$r_{t,\varphi}$	A realization of $\tilde{R}_{t,u}$
$x_{t,u}^{\varphi_{t,u}}$	Normalized bandwidth allocated to user u in time slot t
$c_{t,u}$	Maximum achievable data rate with stochastic form $\tilde{c}_{t,u}$
$f_{C_{t,u}}(c)$	PDF of $\tilde{c}_{t,u}$
$T_{t,u}$	Indicate whether user u completes transmission in time slot t
$J_{t,u}$	Indicate whether user u is scheduled in time slot t
$s_{t,u}^{\varphi}$	Indicate whether user u accesses to BS φ in time slot t
η_u	Weight of user u
$b_{t,u}$	Fraction of data to be transmitted to user u in time slot t
$\mathbf{s}_{t,u}$	State of user u at time slot t
$a_{t,u}$	Action of user u at time slot t
$b'_{t,u}$	Fraction of data transmitted to user u till time slot t
${x_{t,u}^{\hat{\varphi}_{t,u}}}'$	Softmax function is applied to it to get $x_{t,u}^{\hat{\varphi}_{t,u}}$
$Q^{\mu}(s_t, a_t)$	Action-value function
$\mu(s_t)$	Policy function
θ^Q	Parameter of OCN
$\theta^{Q'}$	Parameter of TCN
θ^{μ_u}	Parameter of OAN of actor u
$\theta^{\mu'_u}$	Parameter of TAN of actor u
$L(\theta^Q)$	Loss function
$\nabla_{\theta^{\mu_u}} J$	Policy gradient for actor u
w	Target network updating rate
J	Expected long term discounted reward

III. SYSTEM MODEL

We consider a two-tier time-slotted downlink orthogonal frequency division access (OFDMA) HetNet consisting of macro BSs (MBS) $\mathcal{N}_1$ and small BSs (SBS) $\mathcal{N}_2$ collocated. We assume that different BSs use different frequency bands. Therefore, there is no interference. Locations of MBSs and SBSs are drawn from homogeneous spatial Poisson point process (SPPP) with density of λ_1 and λ_2, respectively. Let $\Phi = \{\varphi | \varphi \in \mathcal{N}_1 \cup \mathcal{N}_2\}$ denote the set of all BSs. Mobile users $\mathcal{U} = \{u | u = 1, 2, \ldots, U\}$ with NRT service request a file of *B* bits. They associate with the BS providing maximum signal to noise ratio (SNR). Considering data rate is a key factor for determining QoS, the minimum data rate requirement c_u^{min} is taken as the QoS requirement. To make it clear, we summarize main notations in Table I.

Users' mobility traces are first predicted. As we utilize PCP to model rate uncertainty in a probabilistic sense, only rate distribution is needed but not the exact realizations. Namely, there is no need to predict the exact values of future channel conditions. A central controller is connected with all BSs to gather historical data for mobility prediction and perform resource allocation.

Frequency bandwidth is reserved at each BS for real-time (RT) service which is non-delay-tolerant and must be served immediately. Only residual frequency bandwidth is available for NRT service which is delay-tolerant and will be queued if there is no sufficient frequency bandwidth. Given users' movements, NRT users may move out of the network's

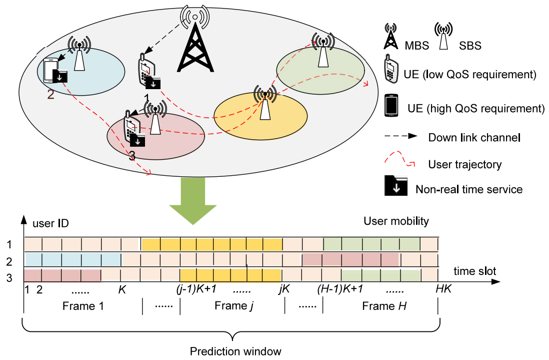


Fig. 1. Overview of mobility model.

coverage before they are scheduled. Hence in time domain, users with higher levels of mobility intensity should be scheduled before those with lower levels. Moreover, resource allocation should fit different QoS requirements. For example, BSs should allocate more frequency bandwidth to those with higher capacity requirements under same channel condition. With the predicted information of user mobility and channel conditions, BSs know when, under what channel conditions and who will compete for resources. Then we can coordinate resource allocation in both time and frequency domains to meet each user's mobility intensity and QoS requirement in large time scale. To further maximize benefits of prediction, robust PRA is designed against prediction uncertainty.

A. Resource Model

Time is divided into slots indexed by t and each with duration of Δ. A set of K time slots is referred to as a frame. The j-th frame is defined as a set $\mathcal{F}_j = \{(j-1)K+1, (j-1)K+2, \dots, jK\}$ of time slots. And the prediction window $\mathcal{H} = \{j, j = 1, 2, \dots, H\}$ consists of H frames. In frequency domain, we explicitly assume that BS φ has pre-reserved certain amount of frequency bandwidth for RT users in time slot t based on a certain resource reservation scheme like [18]. And we denote the residual frequency bandwidth for NRT users as $R'_{t,\varphi}$.

B. Mobility Model

An overview of the mobility model is given in Fig. 1. Position of user u at time slot t is represented by the index of its serving BS denoted by $\varphi_{t,u}$. Thus within a prediction window, the mobility trace $(\varphi_{t,u}, t = 1, 2, \dots, HK)$ of user u is a sequence of time-stamped serving BSs of user u. Mobility of user u can be denoted as a matrix $\mathbf{L}_u = (\mathbf{l}_u^i, i = 1, 2, \dots, l_u)$, where $\mathbf{l}_u^i = (\varphi_u^i, I(\varphi_u^i), \tau(\varphi_u^i))$ is a triple with φ_u^i being the i-th serving BS of user u, which starts to serve the user at time slot $I(\varphi_u^i)$, and $\tau(\varphi_u^i)$ being the residence time slots under BS φ_u^i, l_u is the number of BSs that user u associates with along its trajectory. In this work, we utilize HMM based spatio-temporal travel sequence prediction in [20] to predict mobility trace for user u with historical data record $\mathbf{L}_u$.

HMM characterized by $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\Gamma})$ is composed of hidden states and observable states. We define hidden states as the user's positions in its mobility trace, and the observable states as the HK time slots in the prediction window. States transition matrix $\mathbf{A}$ consists of state transition probabilities $p(\varphi_{t+1,u}|\varphi_{t,u})$ among hidden states. Confusion matrix $\mathbf{B}$ consists of emission probabilities $p(t|\varphi_{t,u})$ that denotes the distribution of observed states that are emitted from each hidden state. $\boldsymbol{\Gamma}$ is consisted of the initial distribution $p(\varphi)$ of hidden states.

Mobility trace prediction is a HMM decoding problem that can be efficiently solved by Viterbi algorithm with obtained $\boldsymbol{\lambda}$. More details can be found in work [20]. The predicted value of $\varphi_{t,u}$ is denoted by $\hat{\varphi}_{t,u}$.

C. Channel Model

In time slot t and $t \in \mathcal{F}_j$, with large scale channel gain $d_{\varphi_{t,u}}^{-\alpha}$ between the user and its serving BS $\varphi_{t,u}$, and small scale channel fading factor $|h_{t,u}|^2 \sim \exp(1)$, the achievable spectral efficiency of user u can be estimated by

$$\gamma_{t,u} = \log_2\left(1 + \frac{P_{\varphi_{t,u}} d_{\varphi_{t,u}}^{-\alpha} |h_{t,u}|^2}{\sigma^2}\right), \tag{1}$$

where $P_{\varphi_{t,u}}$ is the transmit power of BS $\varphi_{t,u}$, $d_{\varphi_{t,u}}$ is the distance between user u and BS $\varphi_{t,u}$, α is the path loss exponent, and σ^2 is the variance of random Gaussian noise. Assume that users change locations over frames. Thus, we have $\varphi_{t,u} = \varphi_{(j-1)K+1,u}$.

IV. MOBILITY-AWARE ROBUST PRA

This work coordinates time slots and frequency bandwidth allocation among multiple mobile NRT users within a prediction window. The goal is to minimize service delay with adaptation to different QoS requirements and mobility intensities. PRA optimization with perfect prediction is first modeled to evaluate the fundamental benefits of proactive algorithm design and provide a performance upper bound. Then PCP is utilized to handle prediction uncertainty and make PRA robust.

A. Problem Formulation for Mobility-Aware PRA with Perfect Prediction (MPRA-Perfect)

In this subsection, we assume that the precisely predicted users' mobility traces and channel gains are known at the beginning of the first time slot.

Let $\mathbf{x}_u = (x_{t,u}^{\varphi_{t,u}}, t = 1, 2, \dots, HK)$ denote the resource allocation vector of user u. $x_{t,u}^{\varphi_{t,u}} \in (0, 1]$ indicates that user u is scheduled in time slot t and occupies $x_{t,u}^{\varphi_{t,u}} R_{t,u}$ amount of frequency bandwidth, where $R_{t,u} = R'_{t,\varphi_{t,u}}$ is the total available frequency bandwidth for user u in time slot t. Otherwise $x_{t,u}^{\varphi_{t,u}} = 0$. For power allocation, we assume that the fraction of power allocated to user u equals to $x_{t,u}^{\varphi_{t,u}}$. The achievable rate of user u in time slot t at BS $\varphi_{t,u}$ is

$$c_{t,u} = R_{t,u} \gamma_{t,u}. \tag{2}$$

In order to model PRA optimization as a convex problem, we introduce a binary vector $\mathbf{T}_u = (T_{t,u}, T_{t,u} \in \{0, 1\}, t = 1, \dots, HK)$ to account for service delay. $T_{t,u} = 1$ indicates that at time slot t, there are still bits remaining to be transmitted for user *u*. Otherwise $T_{t,u} = 0$. Thus, the service delay of user *u* is $\|\mathbf{T}_u\|_1$. To this end, constraint $T_{t,u} \geq \frac{\mathbf{I}_t^{\mathrm{T}}\mathbf{x}_u}{HK}$ must be satisfied, where $\mathbf{I}_t = [\underbrace{0,\ldots,0}_{t-1},1,\ldots,1]^T$.

Define an association indicator $s_{t,u}^{\varphi} \in \{0,1\}$. $s_{t,u}^{\varphi}=1$ indicates that at time slot *t*, user *u* is associated with BS φ. Otherwise $s_{t,u}^{\varphi} = 0$. With the knowledge of user mobility and channel conditions, $c_{t,u}$ and $s_{t,u}^{\varphi}$ are known.

For the sake of adaptation to different users' QoS requirements and mobility intensities indicated by the average cell residence time $\bar{\tau}_u$, we design a weight for user *u*

$$\eta_u = \frac{e^{c_u^{\min}}}{\bar{\tau}_u}. \tag{3}$$

Then we have the PRA optimization problem in (4). The objective is to minimize weighted sum of all users' service delay. Constraint *C*3 is the frequency bandwidth restrict at each BS. Constraint *C*4 ensures the completion of *B* bits data transmission. Constraint *C*5 indicates that the QoS requirement of user *u* must be guaranteed when it is scheduled, where $J_{t,u} = \mathbf{1}\{x_{t,u} > 0\}$ with $\mathbf{1}\{\}$ being an indicator function. We ignore the mobility constraint $\|\mathbf{T}_u\|_1 \leq \sum_i \tau(\varphi_u^i)$. It indicates that data transmission should be completed before the user leaving the network's coverage. Actually, if constraints *C*4 and *C*5 are satisfied, the mobility constraint will be guaranteed. This is because, the user terminates its transmission at time slot $\|\mathbf{T}_u\|_1$ if *C*4 is satisfied, which means that at time slot $\|\mathbf{T}_u\|_1$ *C*5 must also be satisfied and the user must be within the network's coverage.

$$\begin{aligned}
&\underset{\mathbf{x}_u,\mathbf{T}_u,\mathbf{J}_u}{arg\min} \sum_{u\in\mathcal{U}} \eta_u \|\mathbf{T}_u\|_1 \\
&s.t. \quad C1: t = 1,2,\ldots,HK, u\in\mathcal{U}, \varphi\in\Phi, \\
&\qquad C2: x_{t,u}^{\varphi_{t,u}} \in [0,1], T_{t,u}\in\{0,1\}, J_{t,u}\in\{0,1\}, \\
&\qquad C3: \sum_{u\in\mathcal{U}} x_{t,u}^{\varphi_{t,u}} s_{t,u}^{\varphi} \leq 1, \\
&\qquad C4: \Delta\sum_{t=1}^{HK} x_{t,u}^{\varphi_{t,u}} c_{t,u} \geq B, \\
&\qquad C5: x_{t,u}^{\varphi_{t,u}} c_{t,u} \geq J_{t,u} c_u^{\min}, \\
&\qquad C6: T_{t,u} \geq \frac{\mathbf{I}_t^T \mathbf{x}_u}{HK}, \\
&\qquad C7: J_{t,u} \geq x_{t,u}.
\end{aligned} \tag{4}$$

Problem (4) is a mixed integer convex problem that can be solved by convex optimization tools, such as CVX. The difficulty of directly solving problem (4) highly increases with the size of the prediction window. To reduce complexity, we decompose problem (4) in the whole prediction window into sub-optimization problem in each frame and then solve the sub-optimization problems in an iterative manner [30]. The procedure is presented in Algorithm 1. Define $\mathbf{Y} = [\mathbf{X},\mathbf{T},\mathbf{J}]$. Let $\mathbf{Y}(j)$ denote variables in frame *j*, namely $\mathbf{Y}(j) = [\mathbf{X}(j),\mathbf{T}(j),\mathbf{J}(j)]$, where $\mathbf{X}(j) = [x_{t,u}^{\varphi_{t,u}}, t\in\mathcal{F}_j, u\in\mathcal{U}]$, $\mathbf{T}(j) = [T_{t,u}, t\in\mathcal{F}_j, u\in\mathcal{U}]$ and $\mathbf{J}(j) = [J_{t,u}, t\in\mathcal{F}_j, u\in\mathcal{U}]$.

Algorithm 1 Iterative Decision for MPRA-Perfect

Initialize: $\mathbf{Y}$
1: **while** $i<$ maximum iteration number **do**
2: $\quad j = H$
3: $\quad$ **while** $j > 0$ **do**
4: $\qquad$ Fix $\mathbf{Y}\left(j'\right), j' \in \mathcal{H}\backslash j$ and minimize the objective function in problem (4) over frame j by CVX
5: $\qquad$ Update $\mathbf{Y}(j)$ with the optimal solution obtained in line 4
6: $\qquad j \leftarrow j-1$
7: $\quad$ **end while**
8: **end while**
Output: $\mathbf{Y}$

As the objective is to minimize service delay, we start with $j = H$ (line 2) and update **Y**(*j*) in inverted time order (line 6). Variables in all frames except frame *j* are fixed when updating **Y**(*j*) (line 4).

B. Problem Formulation for MRPRA With PCP

We use PCP to tackle prediction uncertainty. In problem (4), the predicted uncertain information includes $s_{t,u}^{\varphi}$, $\gamma_{t,u}$, $\varphi_{t,u}$ and $R_{t,u}$. They are represented by stochastic variables $\tilde{s}_{t,u}^{\varphi}$, $\tilde{\gamma}_{t,u}$, $\tilde{\varphi}_{t,u}$ and $\tilde{R}_{t,u}$, respectively. The meaning of problem (4) is not clearly defined without knowing a realization of the stochastic variables. Thus problem (4) is revised to a deterministic equivalent form with PCP. The stochastic achievable rate is represented by a random variable $\tilde{c}_{t,u} = \tilde{R}_{t,u}\tilde{\gamma}_{t,u}$. It translates prediction uncertainty of mobility and channel conditions to rate uncertainty. The randomness may cause violations in constraints $C3-C5$. By rewriting $C3-C5$ in a probabilistic form, problem (4) is equivalently transferred to (5), where *C*8, *C*9 and *C*10 are transferred from *C*3, *C*4 and *C*5, respectively. *C*8 means that $\sum_{u\in\mathcal{U}} x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{s}_{t,u}^{\varphi} \leq 1$ must be satisfied for any given $\tilde{s}_{t,u}^{\varphi}$. *C*9 guarantees that the probability of user *u* failing to receive *B* bits data doesn't exceed $\varepsilon_1 \in [0,1]$. Similarly, *C*10 guarantees that the probability of QoS requirement violation is no greater than $\varepsilon_2 \in [0,1]$.

$$\begin{aligned}
&\underset{\mathbf{x}_u,\mathbf{J}_u,\mathbf{T}_u}{arg\min} \sum_{u\in\mathcal{U}} \eta_u \|\mathbf{T}_u\|_1 \\
&s.t. \quad C1, C2, C6, C7, \\
&\qquad C8: \mathbb{P}\left\{\sum_{u\in\mathcal{U}} x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{s}_{t,u}^{\varphi} \leq 1\right\} = 1, \\
&\qquad C9: \mathbb{P}\left\{\sum_{t=1}^{HK} \Delta x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{c}_{t,u} < B\right\} \leq \varepsilon_1, \\
&\qquad C10: \mathbb{P}\left\{x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{c}_{t,u} < J_{t,u} c_u^{\min}\right\} \leq \varepsilon_2,
\end{aligned} \tag{5}$$

Proposition 1: The necessary and sufficient condition of constraint *C*4 is constraints $C11: \Delta x_{t,u}^{\hat{\varphi}_{t,u}} c_{t,u} \geq b_{t,u} B$ and $C12: \sum_{t=1}^{HK} b_{t,u} = 1$, where $b_{t,u} \in [0,1]$ represents the fraction of data at least to be transmitted to user *u* in time slot *t*.

Proof: See Appendix A. ■

The probability in $C9$ is hard to derive as it involves cumulative sum of multiple i.i.d. random variables. Proposition1 shows that $C4$ can be decomposed into $C11$ and $C12$. Then $C9$ is replaced by $C13$: $\mathbb{P}\{\Delta x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{c}_{t,u} < b_{t,u} B\} \leq \varepsilon_1$. Define $\mathbf{b}_u = (b_{t,u}, t = 1, 2, \ldots, HK)$. Then problem (5) is modified as

$$\begin{aligned} &\underset{\mathbf{x}_u, \mathbf{J}_u, \mathbf{T}_u, \mathbf{b}_u}{arg\min} \sum_{u \in \mathcal{U}} \eta_u \|\mathbf{T}_u\|_1 \\ &s.t. \quad C1, C2, C6 - C8, C10, C12, C13. \end{aligned} \tag{6}$$

Lemma 1: The PDF $f_\gamma(\xi)$ of the achievable spectral efficiency $\tilde{\gamma}_{t,u}$ is

$$f_\gamma(\xi) = \sum_{k=1}^{2} \frac{2\sigma^2 \pi \lambda_k 2^\xi \ln 2}{\alpha P_k} \int_0^\infty y^{2/\alpha} e^{-y\left(2^\xi - 1\right)\sigma^2/P_k - B_k y^{2/\alpha}} dy, \tag{7}$$

where $B_k = \pi \sum_{j=1}^{2} \lambda_j (\frac{P_j}{P_k})^{2/\alpha}$, P_j and P_k are transmit power of BSs in tier j and tier k, respectively.

Proof: See Appendix B. ■

Lemma 2: The probability mass function (PMF) $p_{t,u}^{\varphi}$ of the total available frequency bandwidth $\tilde{R}_{t,u}$ of user u in time slot t is

$$p_{t,u}^{\varphi} = \frac{\mathbb{P}\left\{t | \tilde{\varphi}_{t,u} = \varphi, R'_{t,\varphi} = r_{t,\varphi}\right\} p(\varphi)}{\sum_{\varphi' \in \Phi} p(\varphi') \mathbb{P}\left\{t | \tilde{\varphi}_{t,u} = \varphi', R'_{t,\varphi'} = r_{t,\varphi'}\right\}}. \tag{8}$$

Proof: See Appendix C. ■

Theorem 1: The PDF $f_{C_{t,u}}(c)$ of the achievable rate $C_{t,u}$ of user u in time slot t is

$$\begin{aligned} f_{C_{t,u}}(c) = &\frac{2\sigma^2 \pi \ln 2}{\alpha} \int_0^\infty y^{2/\alpha} \sum_{r_{t,\varphi} > 0} \frac{2^{c/r_{t,\varphi}} p_{t,u}^{\varphi}}{r_{t,\varphi}} \\ &\times \sum_{k=1}^{2} \frac{\lambda_k}{P_k} \mathrm{e}^{-y\left(2^{c/r_{t,\varphi}} - 1\right)\sigma^2/P_k - B_k y^{2/\alpha}} dy, \end{aligned} \tag{9}$$

where φ is the BS that has $r_{t,\varphi}$ residual frequency bandwidth in time slot t.

Proof: See Appendix D. ■

With Theorem 1, probabilities in $C10$ and $C13$ are deduced as (10) and (11), as shown at the bottom of the next page, respectively.

V. Deep Reinforcement Learning for Mobility-Aware Robust PRA

It can be found that problem (6) is mixed integer and non-convex by substituting (10) and (11) into (6). The problem space sharply increases with size of the prediction window. Furthermore, robust PRA is performed under complex environment. DRL is utilized to handle the above difficulties. Taking advantage of the feature of sequential decision making in DRL, problem of robust PRA can be solved slot by slot. Namely, it only needs to determine $x_{t,u}^{\hat{\varphi}_{t,u}}$ instead of $\mathbf{x}_u$ for each user at each decision epoch t. The agent learns to complete data transmission as soon as possible by maximizing the long term reward with the properly designed reward function.

A. Deep Deterministic Policy Gradient Algorithm

Problem (6) can be modeled as a discrete time MDP with continuous state space $\mathcal{S}$ and action space $\mathcal{A}$. Since the state transition probability and the expected rewards for all states are often unknown, a model-free DRL algorithm DDPG [27] which can tackle continuous actions and states is introduced.

Let the central controller be the agent performing DDPG and each time slot t in the prediction window be a decision epoch. At decision epoch t, the agent takes an action $a_t \in \mathcal{A}$ according to the deterministic policy $\mu : \mathcal{S} \to \mathcal{A}$ that maps state s_t to a specific action a_t after observing current state $s_t \in \mathcal{S}$. Then it receives a reward $r(s_t, a_t)$ and experiences state transition to s_{t+1}. The agent aims to learn a policy that maximizes the expected long term discounted reward $J = \mathbb{E}_{s_t}[\sum_{t=0}^{\infty} \phi^t r(s_t, a_t)]$, where ϕ is a discount factor.

DDPG is composed of an actor and a critic. Role of the actor is to maintain a policy function μ that outputs continuous action given the observed state. Role of the critic is to maintain an action-value function that describes the long term expected feedback after taking action a_t in state s_t following policy μ. It is used to criticize the current policy and defined by

$$Q^\mu(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r(s_t, a_t) + \phi Q^\mu(s_{t+1}, \mu(s_{t+1}))], \tag{12}$$

where $Q^\mu(s_{t+1}, \mu(s_{t+1}))$ and $\mu(s_{t+1})$ are target values of the action-value function and policy function, respectively.

1) Critic: The critic utilizes a DNN with parameter θ^Q, called online critic network (OCN) $Q^\mu(s_t, \mu(s_t|\theta^\mu)|\theta^Q)$, to estimate the action-value function. OCN is trained to make correct criticism on the current policy by minimizing the loss

$$L\left(\theta^Q\right) = \mathbb{E}_{s_t}\left[\left(Q^\mu\left(s_t, \mu(s_t|\theta^\mu)|\theta^Q\right) - y_t\right)^2\right], \tag{13}$$

with gradient descent algorithm (GDA), where $y_t = r(s_t, a_t) + \phi Q^{\mu'}(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$. The loss function tells how bad the action-value function is estimated compared to the expected. To calculate y_t, the critic uses a separate DNN with parameter $\theta^{Q'}$, called target critic network (TCN) $Q^{\mu'}(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$ to get the target value $Q^\mu(s_{t+1}, \mu(s_{t+1}))$ in (12). TCN has the same structure as OCN and is updated by

$$\theta^{Q'} \leftarrow w\theta^Q + (1 - w)\theta^{Q'}, \tag{14}$$

with updating rate $w \ll 1$.

2) Actor: The policy function μ is estimated by a DNN with parameter θ^μ called online actor network (OAN) $\mu(s_t|\theta^\mu)$. OAN is trained with gradient ascent algorithm (GAA). And the critic guides the training by providing its criticism $\nabla_\mu Q^\mu(s_t, \mu(s_t|\theta^\mu)|\theta^Q)$ on the current policy to the policy gradient

$$\nabla_{\theta^\mu} J = \mathbb{E}_{s_t}\left[\nabla_\mu Q^\mu\left(s_t, \mu(s_t|\theta^\mu)|\theta^Q\right) \nabla_{\theta^\mu} \mu(s_t|\theta^\mu)\right]. \tag{15}$$

By applying $\nabla_{\theta^\mu} J$ as the gradient to GAA, parameter θ^μ is updated in a direction that would maximize J.

The actor also maintains a target actor network (TAN) $\mu'(s_{t+1}|\theta^{\mu'})$ with parameter $\theta^{\mu'}$ to calculate the target value $\mu(s_{t+1})$ in (12). TAN is a copy of OAN and is updated with the same rule in (14).

3) Training Process: The agent stores transition $(s_t, a_t, r(s_t, a_t), s_{t+1})$ notated by m in a replay memory (RM) $\mathcal{M}$. In this work we adopt the prioritized sampling strategy in [28] to improve performance of DDPG. Each transition has a sampling probability defined by

$$q_m = \frac{1/rank(m)}{\sum\limits_{m' \in \mathcal{M}} 1/rank(m)}, \tag{16}$$

where function *rank*(m) gives the rank of transition m in $\mathcal{M}$ based on the loss value $L_m(\theta^Q)$ calculated by (13) with transition m.

In each training episode, a mini-batch of transitions $\mathcal{D}$ are sampled from $\mathcal{M}$ based on their sampling probabilities. OCN is first trained by minimizing the loss $\frac{1}{|\mathcal{D}|} \sum\limits_{m' \in \mathcal{D}} W_{m'} L_{m'}(\theta^Q)$, where $W_{m'} = [1/(|\mathcal{M}| q_{m'})]^\beta$ is the importance sampling weight with parameter $\beta \in [0, 1]$.

Then OCN calculates the action-value function with sampled transitions to get the criticism $\nabla_\mu Q^\mu(s_t, \mu(s_t|\theta^\mu)|\theta^Q)$. After that, OAN is updated using the sampled policy gradient $\frac{1}{|\mathcal{D}|} \sum\limits_{m' \in \mathcal{D}} \nabla_{\theta^\mu}^{m'} J$, where $\nabla_{\theta^\mu}^{m'} J$ is the policy gradient calculated with sample m'. Finally, parameters of TAN and TCN are updated with (14). The whole process repeats till convergence.

B. Multi-Actor DDPG Based MRPRA Decision Making

In order to flexibly coordinate resource allocation among multiple users over time horizon, we extend DDPG with only one actor to multi-actor DDPG which works in a way of distributed acting and centralized criticizing. It uses multiple actors and each stands for a user to learn its own policy. This is motivated by the idea of multi-task DDPG proposed in [29]. Here and after, terms ‘user’ and ‘actor’ are used interchangeably. Users who complete their data transmission earlier won’t take actions any more but wait for the others. It is difficult to control one single actor to perform such process. Thus it’s necessary to use multiple actors that can flexibly control their own actions.

The framework of multi-actor DDPG is show in Fig. 2. At decision epoch t, each actor takes an action $a_{t,u}$ after observing the global state $\mathbf{s}_t$. Then each actor receives an individual reward $r_u(\mathbf{s}_t, \mathbf{a}_t)$ under action profile $\mathbf{a}_t = (a_{t,1}, a_{t,2}, \ldots, a_{t,U})$. After processing all actors’ individual rewards, the global reward $r(\mathbf{s}_t, \mathbf{a}_t)$ is obtained and stored in a RM together with $\mathbf{s}_t$, $\mathbf{a}_t$ and new global state $\mathbf{s}_{t+1}$. Then the critic takes a batch of samples from the RM to evaluate each actor’s policy. And actors update their policies based on the critic’s evaluation.

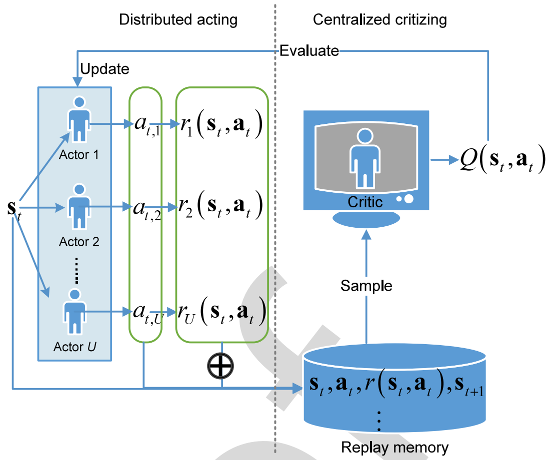


Fig. 2. Multi-actor DDPG based MRPRA decision making framework.

The global state space $\mathcal{S}$ is composed of all users’ state spaces $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \mathcal{S}_U$. Define the state of user u at decision epoch t as $\mathbf{s}_{t,u} = (b'_{t,u}, \hat{\varphi}_{t,u}, p_{t,u}^{\varphi}, r_{t,\varphi}, \varphi \in \Phi)$, where $b'_{t,u} \in [0, 1]$ is the fraction of data transmitted to user u till decision epoch t. We can assume that each user has a buffer of size 1, and the buffer state is $b'_{t,u}$ indicating the data amount in the buffer. The corresponding global state at decision epoch t is $\mathbf{s}_t = (\mathbf{s}_{t,u}, u \in \mathcal{U})$.

We define action taken by user u at decision epoch t as $a_{t,u} = {x_{t,u}^{\hat{\varphi}_{t,u}}}'$. SoftMax function is applied to compute $x_{t,u}^{\hat{\varphi}_{t,u}}$, that is $x_{t,u}^{\hat{\varphi}_{t,u}} = \frac{{x_{t,u}^{\hat{\varphi}_{t,u}}}'}{\sum\limits_{u' \in \mathcal{U}} 1\{\hat{\varphi}_{t,u'} = \hat{\varphi}_{t,u}\} {x_{t,u'}^{\hat{\varphi}_{t,u'}}}'}$, which catches constraint $C8$. The actions profile of all users at decision epoch t is $\mathbf{a}_t = (a_{t,u}, u \in \mathcal{U})$.

Theorem 2: The agent only needs to determine $\mathbf{X}$. And the optimal solution of $b_{t,u}$ is $b_{t,u}^* = \frac{\bar{\psi}_{t,u} \Delta x_{t,u}^{\hat{\varphi}_{t,u}}}{B}$ when $x_{t,u}^{\hat{\varphi}_{t,u}}$ is fixed, where $\int_0^{\bar{\psi}_{t,u}} f_{C_{t,u}}(c) dc = \varepsilon_1$.

Proof: See Appendix E. ■

After user u takes an action $a_{t,u}$ under global state $\mathbf{s}_t$ it receives an individual reward $r_u(\mathbf{s}_t, \mathbf{a}_t)$ and a global reward $r(\mathbf{s}_t, \mathbf{a}_t)$ that the agent aims to maximize. The reward function should be designed carefully otherwise the agent hardly learns anything. The agent aims to learn policies that minimizes weighted sum of service delay under constraint $C10$. So the reward function should be characterized by, a) it can capture violation of $C10$, b) it can coordinate resource allocation among users according to the weight η_u, c) it can stimulate the agent to reduce service delay. To this end, the individual reward function and global reward function are defined

$$\mathbb{P}\left\{x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{c}_{t,u} < J_{t,u} c_u^{\min}\right\} = \frac{2\sigma^2 \pi \ln 2}{\alpha} \int_0^{\frac{J_{t,u} c_u^{\min}}{x_{t,u}^{\hat{\varphi}_{t,u}}}} \int_0^\infty y^{2/\alpha} \sum_{r_{t,\varphi} > 0} \frac{2^{c/r_{t,\varphi}} p_{t,u}^{\varphi}}{r_{t,\varphi}} \sum_{j=1}^{2} \frac{\lambda_j}{P_j} e^{-y\left(2^{c/r_{t,\varphi}} - 1\right)\sigma^2/P_j - B_j y^{2/\alpha}} dy dc \tag{10}$$

$$\mathbb{P}\left\{\Delta x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{c}_{t,u} < b_{t,u} B\right\} = \frac{2\sigma^2 \pi \ln 2}{\alpha} \int_0^{\frac{b_{t,u} B}{\Delta x_{t,u}^{\hat{\varphi}_{t,u}}}} \int_0^\infty y^{2/\alpha} \sum_{r_{t,\varphi} > 0} \frac{2^{c/r_{t,\varphi}} p_{t,u}^{\varphi}}{r_{t,\varphi}} \sum_{j=1}^{2} \frac{\lambda_j}{P_j} e^{-y\left(2^{c/r_{t,\varphi}} - 1\right)\sigma^2/P_j - B_j y^{2/\alpha}} dy dc \tag{11}$$

in (17) and (18), respectively.

$$r_u(\mathbf{s}_t, \mathbf{a}_t) = \eta_u\big(b'_{t+1,u} - 1\big) + p_1, \tag{17}$$

$$r(\mathbf{s}_t, \mathbf{a}_t) = \sum_{u \in \mathcal{U}} r_u(\mathbf{s}_t, \mathbf{a}_t) + \rho W. \tag{18}$$

Define function $g^-(x) = \begin{cases} 0, & x \geq 0 \\ x, & x < 0 \end{cases}$. In (17), $p_1 = g^-(\varepsilon_2 - \mathbb{P}\{x_{t,u}^{\varphi_{t,u}} \tilde{c}_{t,u} < J_{t,u} c_u^{\min}\})$ is the penalty of violating $C10$. The term $\eta_u(b'_{t+1,u} - 1)$ in (17) indicates the weighted data amount remaining to download. The agent tends to preferentially serve users with large η_u to maximize (18). In order to stimulate the agent to shorten service delay, we award bonus ρW to it, where $\rho = \mathbf{1}\{\forall b'_{t+1,u} = 1, u \in \mathcal{U}\}$ is a terminal indicator.

After receiving a reward, each user's state transits to $\mathbf{s}_{t+1,u} = \begin{cases} (\mathbf{s}_{1,u}, u \in \mathcal{U}), & \text{if } \rho = 1 \\ (b'_{t+1,u}, \hat{\varphi}_{t+1,u}, p^{\varphi}_{t+1,u}, r_{t+1,\varphi}, \varphi \in \Phi), & \text{else} \end{cases}$, where $b'_{t+1,u} = \begin{cases} \min(b'_{t,u} + b^*_{t,u}, 1), & \text{if } C10 \text{ is satisfied} \\ b'_{t,u}, & \text{otherwise} \end{cases}$. Namely, user u fails to download any bits at decision epoch t if $C10$ is violated. It's noteworthy that when all users finish data transmission, we set the new global state as the initial one for stable state transition.

Define policies profile $\boldsymbol{\mu} = (\mu_u(\mathbf{s}_t|\theta^{\mu_u}), u \in \mathcal{U})$. Correspondingly, the action-value function is $Q^{\boldsymbol{\mu}}(\mathbf{s}_t, \boldsymbol{\mu}|\theta^Q)$, and y_t is

$$y_t = r(\mathbf{s}_t, \mathbf{a}_t) + \phi Q^{\boldsymbol{\mu}'}\Big(\mathbf{s}_{t+1}, \boldsymbol{\mu}'|\theta^{Q'}\Big), \tag{19}$$

where $\boldsymbol{\mu}' = (\mu'_u(\mathbf{s}_{t+1}|\theta^{\mu'_u}), u \in \mathcal{U})$.

For actor u, the policy gradient is

$$\nabla_{\theta^{\mu_u}} J = \mathbb{E}_{\mathbf{s_t}}[\nabla_{\mu_u} Q^{\boldsymbol{\mu}}\Big(\mathbf{s}_t, \boldsymbol{\mu}|\theta^Q\Big) \nabla_{\theta^{\mu_u}} \mu_u(\mathbf{s}_t|\theta^{\mu_u})]. \tag{20}$$

The process of multi-actor DDPG based MRPRA decision making is given in Algorithm 2.

Line 5–line 11: Whether data transmission is finished is checked for each user at each decision epoch. If user u finishes transmission, it does nothing but waits for other users. Otherwise, it outputs current action. Line 12–line 17: The agent observes global reward and new global state after all users take actions. Then it saves transition $(\mathbf{s}_t, a_t, r(\mathbf{s}_t, a_t), \mathbf{s}_{t+1})$ in RM $\mathcal{M}$. Line 18–line 26: The agent trains its OCN, TCN, OANs and TANs based on the training process in Section V-A3.

After all users finish data transmission, the current training episode terminates and the next training episode starts.

C. Resource Allocation

The output action profile $(\mathbf{a}_t, t = 1, 2, \ldots, HK)$ of Algorithm 2 gives the resource allocation plan in a given prediction window. When time slot t comes, the central controller already knows $\{\varphi_{t,u}, u \in \mathcal{U}\}$, it informs BS $\varphi_{t,u}$ to schedule user u with $\frac{x_{t,u}^{\varphi_{t,u}{}'}}{\sum_{u' \in \mathcal{U}} 1\{\varphi_{t,u'} = \varphi_{t,u}\} x_{t,u'}^{\varphi_{t,u'}{}'}} R_{t,u}$ amount of frequency bandwidth. It's noteworthy that user u gets resources from its actual serving BS $\varphi_{t,u}$ instead of the predicted one.

Algorithm 2 Multi-Actor DDPG Based MRPRA Decision Making

Initialize: $\mathbf{s}_{1,u} = \Big(0, \hat{\varphi}_{1,u}, p^{\varphi}_{1,u}, r_{1,\varphi}, \varphi = 1, 2, .., |\Phi|\Big)$, θ^Q, θ^{μ_u}, $\theta^{\mu'_u} \leftarrow \theta^{\mu_u}$, $\theta^{Q'} \leftarrow \theta^Q$, $u \in \mathcal{U}$, replay memory $\mathcal{M}$
Input: maximum training episode $E_{\max}$, size of prediction window HK, mini-batch size D
1: **while** episode $< E_{\max}$ **do**
2: Initialize a random process $\mathcal{Z}$ for action exploration
3: $\rho \leftarrow 0, t \leftarrow 1$
4: **while** $\rho = 0$ and $t \leq HK$ **do**
5: **for** $u = 1 : U$ **do**
6: **if** actor u finishes data transmission **then**
7: Set $a_{t,u} = 0$, $r_u(\mathbf{s}_t, \mathbf{a}_t) = 0$, $b'_{t,u} = 1$
8: **else**
9: Select action $a_{t,u} = \mu_u\big(\mathbf{s}_t|\theta^{\mu_u}\big) + \mathcal{Z}_t$
10: **end if**
11: **end for**
12: **for** $u = 1 : U$ **do**
13: Observe reward $r_u(\mathbf{s}_t, \mathbf{a}_t)$ and new state $\mathbf{s}_{t+1,u}$
14: **end for**
15: $\rho \leftarrow \mathbf{1}\Big\{\forall b'_{t+1,u} = 1, u \in \mathcal{U}\Big\}$
16: Observe global reward $r(\mathbf{s}_t, \mathbf{a}_t)$ and new global state $\mathbf{s}_{t+1}$
17: Store transition $\big(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1}\big)$ in $\mathcal{M}$
18: Sample transitions $\mathcal{D}$ from $\mathcal{M}$ according to q_m
19: Compute y_t according to (19)
20: Update parameter θ^Q by minimizing the loss $\frac{1}{D} \sum_{m' \in \mathcal{D}} W_{m'} L_{m'}\Big(\theta^Q\Big)$
21: Update $L_{m'}\Big(\theta^Q\Big)$ and $rank\Big(m'\Big)$, $m' \in \mathcal{D}$
22: Update parameter $\theta^{Q'}$ according to (14)
23: **for** $u = 1 : U$ **do**
24: Update parameter θ^{μ_u} with $\frac{1}{D} \sum_{m' \in \mathcal{D}} \nabla^{m'}_{\theta^{\mu_u}} J$
25: Update parameter $\theta^{\mu'_u}$ according to (14)
26: **end for**
27: $t \leftarrow t + 1$
28: **end while**
29: **end while**
Output: $\mathbf{X}$

This avoids wasting resources if user u is scheduled at BS $\hat{\varphi}_{t,u}$ but $\hat{\varphi}_{t,u} \neq \varphi_{t,u}$. If the actually transmitted data amount of user u is less than B bits, it will be scheduled with FS after time slot $\max_u \|\mathbf{T}_u\|_1$ to transmit the rest data.

VI. Simulations and Analysis

We evaluate performance of the proposed mobility-aware robust PRA method via extensive simulations. All simulation parameters, unless stated otherwise, are listed in Table II. The reactive resource allocation scheme FS is introduced for performance comparison to observe the benefit of proactive algorithm design. MPRA-perfect serves as the performance upper bound. We also simulate mobility-aware non-robust PRA (MPRA-non-robust) method to validate the robustness of MRPRA. The only difference between MPRA-non-robust and MPRA-perfect is that the former applies the imperfectly predicted mobility traces to problem (4). Successful scheduling probability (SSP) and average service delay (ASD) are taken as the performance metrics. SSP is the probability that the

TABLE II
SIMULATION PARAMETER SETTING

Parameter	Value
Radius of simulation region	100 m
Density of MBSs (λ_1)	$2 \cdot 10^{-5}$
Density of SBSs (λ_2)	$5 \cdot 10^{-5}$
Transmit power of MBSs	43 dBm
Transmit power of SBSs	33 dBm
Number of users (U)	5
Noise variance (σ^2)	-106 dBm
Size of prediction window (H)	10 frames
Size of frame (K)	100 time slots
Duration of each time slot (Δ)	10 ms
Memory level of GMM model (θ)	0.5
Mean of velocity ($\bar{v}$)	$\{2,3,4,5,6\}$ m/s
Standard variation of velocity (δ)	0.2
QoS requirement ($c_u^{\min}$)	$\{1,2,3,4,5,6\}$ Mbps
Violation probability in constraint $C13$ (ε_1)	0.1
Violation probability in constraint $C10$ (ε_2)	0.01
Maximum training episode (E_{max})	60
Mini–batch size (D)	32
Learning rate of critic	10^{-3}
Learning rate of actors	10^{-3}
Parameter of importance sampling weight (β)	0.5

user completes B bits data transmission within the prediction window.

Consider a circle simulation region. Residual frequency bandwidth at each BS is Poisson distributed with parameter λ_R. The initial locations of users are uniformly distributed. The moving direction at each frame is uniformly drawn from $[-\pi, \pi]$. The velocity is updated according to Gauss–Markov mobility (GMM) model $v_{j+1} = \theta v_j + (1-\theta)\bar{v} + \delta\sqrt{1-\theta^2}\phi$ [32], where v_j is the velocity at frame j, v_{j+1} is the velocity at the next frame $j+1$, $\theta \in [0, 1]$ indicates the memory level, $\bar{v}$ and δ are mean and standard variation of velocity, ϕ is Gaussian process with zero mean and unit variance. For mobility prediction, we generate 100 trajectories for each user as historical data. And the 101-th trajectory as the real mobility trace. Mobility prediction accuracy is defined as the ratio between the numbers of correctly predicted locations over the total number of locations. HMM achieves 72.72% prediction accuracy.

For TANs and OANs, we use sigmoid (i.e., $y = \frac{1}{1+e^{-x}}$) as the activation function in the output layers to limit output actions to [0, 1]. For TCN and OCN, no activation function is used in the output layers.

We set $B = 500Mbit$, $\lambda_R = 8MHz$ and study convergence properties of Algorithm 1 and Algorithm 2. Fig. 3 shows that Algorithm 1 converges to an optimal solution after iteration 70. Fig. 4 gives the convergence property of Algorithm 2 under different learning rates of actors with the critic's learning rate being fixed to 10^{-3}. It can be found that Algorithm 2 converges under all the learning rate settings. The agent learns the best policy with the learning rate being 10^{-3}. And the performance cannot be improved either with the learning rate increasing to 10^{-2} or decreasing to 10^{-4}. So the actors' learning rate should be chosen properly, neither too large nor too small. Otherwise the agent cannot learn an optimal policy.

The purpose of Fig. 5 is to validate rate distribution derived in Theorem 1. In this simulation, a typical user moves from the origin. Radius of the simulation region is set to 1000 m. Rate CDF is computed for each time slot and the results are averaged over the prediction window. We run simulation 10^5 times. It's shown that the analytic curve obtained from Theorem 1 is in quite good agreement with the simulated one and thus Theorem 1 is validated.

Under different average residual frequency bandwidth, Fig. 6 compares SSP and ASD for different violation probability ε_1 in constraint $C13$. We set $B = 500Mbit$. SSP degrades with ε_1 when $\lambda_R = 3MHz$ and it becomes less sensitive to ε_1 values with λ_R increasing. ASD slightly grows with ε_1 under all the λ_R values. In a whole, performance of the proposed approach is degraded by large ε_1 values. This is because the transmitted data amount $b^*_{t,u}$ in each time slot grows with ε_1.

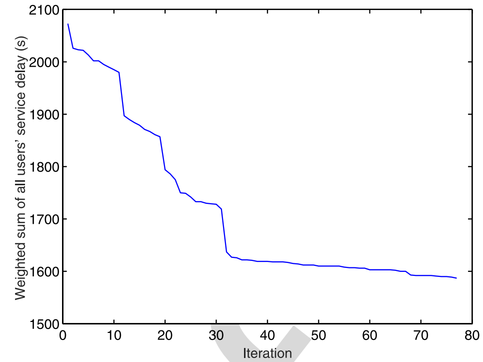


Fig. 3. Convergence property of Algorithm 1.

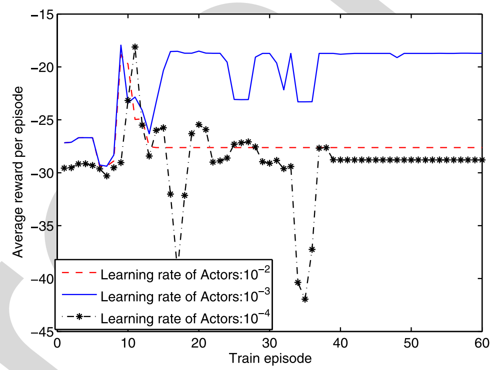


Fig. 4. Convergence property of Algorithm 2 under different learning rate of actors.

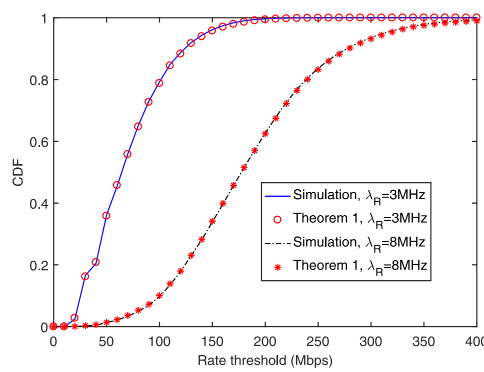


Fig. 5. Comparison of rate distribution obtained from Theorem 1 and simulation.

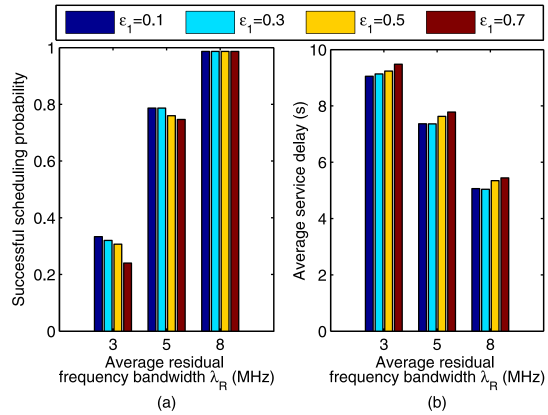


Fig. 6. Successful scheduling probability and average service delay for different ε_1 with different λ_R values.

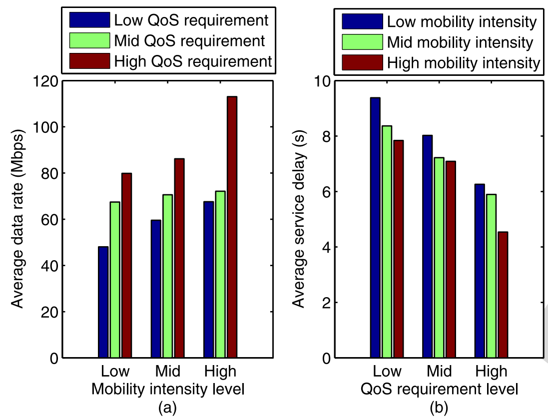


Fig. 7. Comparison of average data rate and average service delay with different level of mobility intensity and QoS requirement for MRPRA.

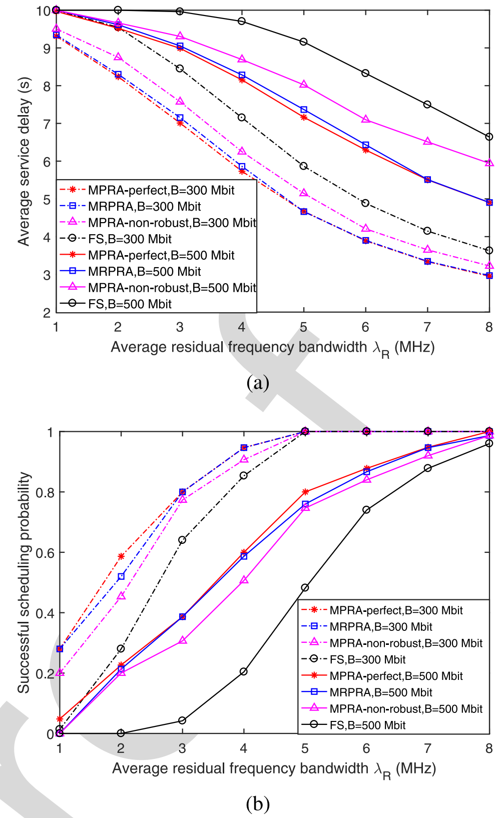


Fig. 8. Comparison of average service delay and successful scheduling probability for different resource allocation approaches under different B values with λ_R varying.

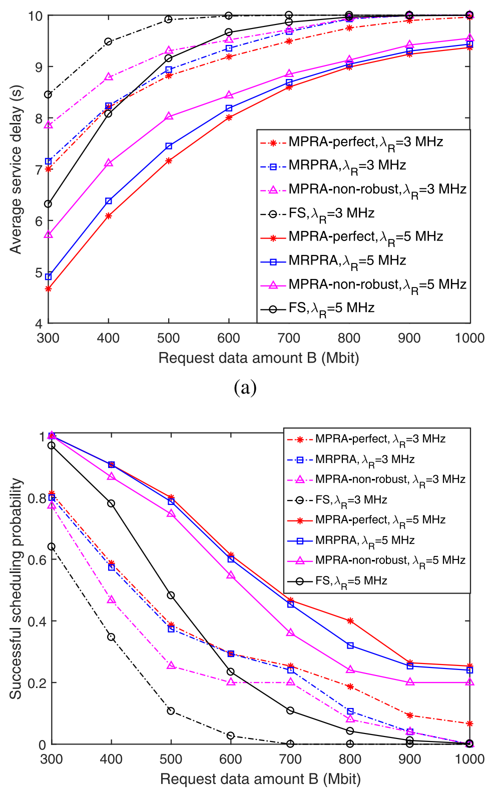
So it takes less time to have $b'_{t,u} = 1$ with larger ε_1 values. Consequently, the agent will terminate transmission for user u and no longer plan to allocate any resources to it. However, the actually transmitted data amount of user u may be less than B bits. The central controller will serve it in a reactive way, which results in large delay and low SSP.

MRPRA aims to adapt to users' mobility intensities and QoS requirements. Fig. 7 studies the adaptiveness. We set $B = 500Mbit$ and $\lambda_R = 5MHz$. QoS requirement is grouped into three levels, low ($c_u^{\min} \in \{1, 2\}$Mbps), mid ($c_u^{\min} \in \{3, 4\}$Mbps), and high ($c_u^{\min} \in \{5, 6\}$Mbps). The higher the level is, the larger the data rate should be. In Fig. 7(a), the average data rate increases with QoS requirement level under each mobility intensity level. This indicates that MRPRA has good adaptiveness to QoS requirements. Mobility intensity is grouped into three levels, low ($\bar{\tau}_u \in [1, 4]$s), mid ($\bar{\tau}_u \in [4, 7]$s), and high ($\bar{\tau}_u \in [7, 10]$s). The higher the level is, the lower the service delay should be, which is exactly the results shown in Fig. 7(b). This indicates quite good adaptiveness of MRPRA to mobility intensity.

With average residual frequency bandwidth λ_R varying, ASD and SSP are compared for different resource allocation approaches under different request data amount in Fig. 8(a) and Fig. 8(b), respectively. MPRA-perfect outperforms the other three approaches. Averagely, 16% improvement in ASD and 232% improvement in SSP are achieved over FS. This indicates that MPRA-perfect can serve much more users and meanwhile shorten service delay compared to FS. Such benefit comes from perfect prediction. Averagely, MPRA-non-robust has 9% performance loss in ASD and 18.5% performance loss in SSP from MPRA-perfect. MRPRA reduces the losses in ASD and SSP to 0.9% and 7.5%, respectively. As a whole, MRPRA performs very close to MPRA-perfect, which indicates that MRPRA guarantees as much data traffic as MPRA-perfect does and shortens service delay.

Performance losses of MRPRA come from imperfectly predicted trajectories. The agent gets wrong figure of interactions among users and coordinates resource allocation improperly. However, with 72.72% mobility prediction accuracy, MRPRA achieves much lower performance losses than MPRA-non-robust, which shows robustness of MRPRA.

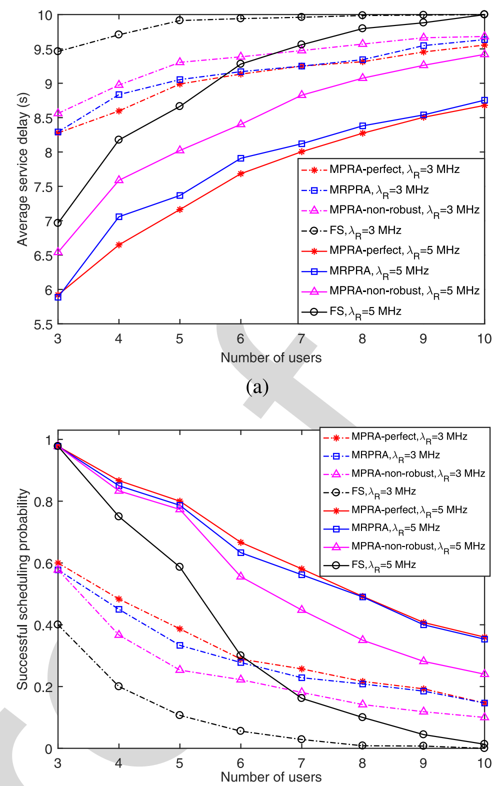
With request data amount B varying, ASD and SSP are compared for different resource allocation approaches in Fig. 9(a) and Fig. 9(b), respectively. The more data traffic is, the more users MPRA-perfect can serve compared to FS. But ASD gets close to that of FS with B increasing. Under such condition, the reactive scheme FS can be activated instead of the proactive approaches for computational simplicity if we ignore SSP. Averagely, MPRA-non-robust has 6% performance loss in ASD and 28% performance loss in SSP from MPRA-perfect. MRPRA reduces the losses in ASD and SSP to 1.5% and 15%, respectively. The actually transmitted data amount is less than



(a)

(b)

Fig. 9. Comparison of average service delay and successful scheduling probability for different resource allocation approaches under different λ_R values with B varying.



(a)

(b)

Fig. 10. Comparison of average service delay and successful scheduling probability for different resource allocation approaches under different λ_R values with number of users varying.

the requested because of biased mobility prediction. The number of under-served users who will be scheduled with FS grows with B. Thus the performance of MRPRA gets close to FS with B increasing and λ_R decreasing. But MRPRA achieves much lower performance losses than MPRA-non-robust, which benefits from the robustness of MRPRA. And it's observed that the performance loss in SSP increases much faster than that in ASD. This indicates that MRPRA tries to guarantee ASD at a cost of dropping some users.

Fig. 10(a) and Fig. 10(b) respectively compare ASD and SSP for different resource allocation methods with the number of users varying. We set $B = 500Mbit$. It shows that gains in both ASD and SSP of MPRA-perfect over FS increase with the number of users and λ_R. Averagely, MPRA-non-robust respectively has 6% and 22% performance losses from MPRA-perfect in ASD and SSP. MRPRA reduces the losses to 1.8% and 3.9%, respectively. And it achieves 10.5% and more than 500% performance gains in ASD and SSP over FS, respectively. On the whole, the performance loss in SSP of MRPRA from MPRA-perfect keeps decreasing when the number of users is greater than 7. However, the ASD loss starts to increase at the tail of x-axis. We therefore conclude that MRPRA tries to guarantee SSP at a cost of delaying some users when the number of users grows large.

In order to validate robustness of the proposed method, we study the impact of mobility prediction error on ASD and SSP in Fig. 11(a) and Fig. 11(b), respectively. We set $B = 500Mbit$ and $\lambda_R = 3MHz$. Fig. 11 shows that the performance losses of

TABLE III
CPU TIME, ASD AND SSP UNDER DIFFERENT K VALUES

Metric	Method \ K	10	20	30	100
CPU time (s)	Algorithm 1	93.19	127.18	159.91	363.5
	CVX solver	347.11	5123.94	9556.55	–
ASD (s)	Algorithm 1	7.42	7.35	7.26	7.41
	CVX solver	7.25	7.21	7.17	–
SSP	Algorithm 1	0.81	0.8	0.81	0.8
	CVX solver	0.85	0.85	0.85	–

MPRA-non-robust in both ASD and SSP from MPRA-perfect grow sharply with prediction error. While losses of MRPRA grow slightly and MRPRA performs very close to MPRA-perfect. Averagely, MPRA-no-robust has 5.3% ASD loss and 33.6% SSP loss, respectively. While, MRPRA holds the losses no greater than 1.5%. As a whole, MRPRA performs much less sensitive to prediction error, which reflects robustness of MRPRA.

As Algorithm 1 obtains a sub-optimal solution for problem (4), we test the CPU time, ASD and SSP to study the computational complexity reduction and performance loss of Algorithm 1. The simulation platform is CPU Intel Core i5-7300HQ. We set $B = 500Mbit$ and $\lambda_R = 5MHz$. Results are shown in Table III.

CPU time of directly solving problem (4) with CVX solver sharply increases with K. When K grows greater than 30, the time cost of CVX solver is unaffordable and no solution can be obtained. By comparison, solving problem (4) with

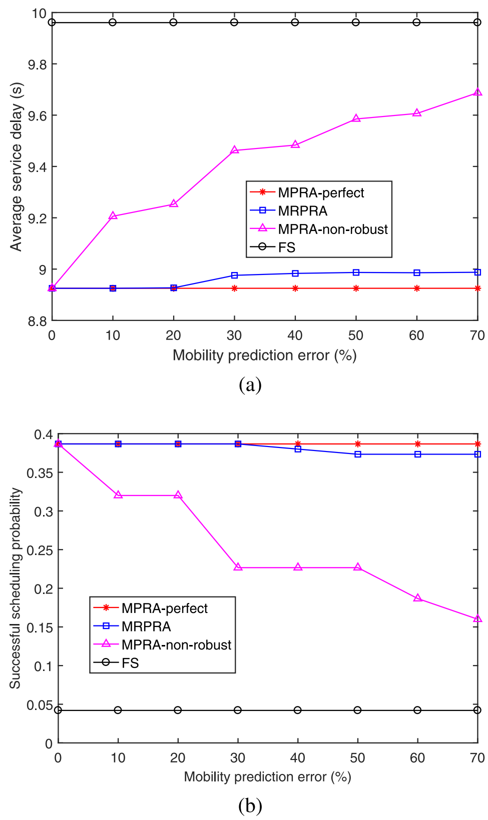


Fig. 11. Comparison of average service delay and successful scheduling probability for different resource allocation approaches with mobility prediction error varying.

TABLE IV
CPU TIME FOR DIFFERENT METHODS

Metric	Method				
	MPRA-perfect	MRPRA (train)	MRPRA (execute)	MPRA-non-robust	FS
CPU time (s)	363.5	2885.26	1.14	426.28	0.03

Algorithm 1 saves more than 73% CPU time. The average performance losses of Algorithm 1 in ASD and SSP from CVX solver are only 2% and 5%, respectively. In a conclusion, Algorithm 1 performs much more efficiently than CVX solver.

We also test CPU time for MRPRA, MPRA-non-robust and FS for computational complexity comparison. We set $B = 500Mbit$ and $\lambda_R = 5MHz$. Results are shown in Table IV.

FS achieves the lowest CPU time. However, Fig. 8–11 show that it performs worst in both terms of ASD and SSP. MRPRA uses multi-actor DDPG algorithm to make decisions. It takes about 8 times as much CPU time as MPRA-perfect and MPRA-non-robust to train multi-actor DDPG. Fortunately, DRL is characterized by "Once trained, run everywhere". Once numbers of inputs and action outputs of multi-actor DDPG are fixed, whenever the environment which contains $R'_{t,\varphi}$, B, users' trajectories, changes, it can immediately make decisions for MRPRA after being well trained. It takes only 1.14 seconds to execute MRPRA. However, MPRA-perfect and MPRA-non-robust need to solve problem (4) again to get the resource allocation plans. Besides, MRPRA performs close to MPRA-perfect compared to FS and MPRA-non-robust. MRPRA is therefore time efficient and robust.

VII. CONCLUSION

In this paper, we have studied how to efficiently exploit prediction and how to handle prediction uncertainty for PRA optimization. Only coarse predicted information is needed. We have modeled PRA with perfect prediction as a mixed integer convex problem to provide a performance upper bound for robust PRA method design. Users' mobility traces are predicted by HMM. To make PRA robust against prediction uncertainty, PCP is utilized to formulate the constraints accommodating the predicted uncertain achievable rate in a probabilistic form. And the rate distribution is derived. We have further modeled robust PRA optimization as a MDP and solved it with our designed multi-actor DDPG algorithm. Simulations demonstrate that the proposed approach has good adaptiveness to users' rate requirements and mobility intensities. The derived PDF of achievable rate is validated. Moreover, it's found that the reactive resource allocation scheme can be performed instead of the proactive one when the available frequency bandwidth is insufficient for computational simplicity. And the proposed method achieves robustness and efficiency.

APPENDIX A
PROOF OF PROPOSITION 1

Proof of Necessity: Might as well divide B into the sum of variables $B_{t,u} \geq 0, t = 1, 2, \ldots, HK$. Then constraint $C4$ can be rewritten by $\sum_{t=1}^{HK} (\Delta x_{t,u}^{\hat{\varphi}_{t,u}} c_{t,u} - B_{t,u}) \geq 0$. As both $x_{t,u}^{\hat{\varphi}_{t,u}}$ and $B_{t,u}$ are no less than zero, we can get $\Delta x_{t,u}^{\hat{\varphi}_{t,u}} c_{t,u} \geq B_t, \forall t$. By normalizing $B_{t,u}$ to $b_{t,u} = \frac{B_{t,u}}{B}$ we can get $\Delta x_{t,u}^{\hat{\varphi}_{t,u}} c_{t,u} \geq b_{t,u} B, \forall t$, which is constraint $C11$. Obviously $\sum_{t=1}^{HK} b_{t,u} = 1$, which is constraint $C12$.

Proof of Sufficiency: Summing both sides of the inequality in $C11$ we can get $\sum_{t=1}^{HK} \Delta x_{t,u}^{\hat{\varphi}_{t,u}} c_{t,u} \geq \sum_{t=1}^{HK} b_{t,u} B$. Substituting $C12$ into the result gives $\sum_{t=1}^{HK} \Delta x_{t,u}^{\hat{\varphi}_{t,u}} c_{t,u} \geq B$, which is constraint $C4$.

APPENDIX B
PROOF OF LEMMA 1

$\tilde{\gamma}_{t,u}$ is i.i.d. among users and time slots, so we omit subscripts of index t and u in the following notations. [31, Lemma 3] gives the PDF $f_{D_k}(d) = \frac{2\pi\lambda_k d}{A_k} e^{-\pi B_k d^2}$ of distance D_k between the user and its serving BS in tier k for max SNR association, where

$$A_k = \left(1 + \frac{\sum\limits_{j=1, j\neq k}^{2} \lambda_j (P_j)^{2/\alpha}}{\lambda_k (P_k)^{2/\alpha}} \right)^{-1}$$

is the probability that the user associates with the k-th tier.

Define $Y_k = D_k^{\alpha}$. The PDF of Y_k is derived from $f_{D_k}(d)$ as $f_{Y_k}(y) = \frac{2\pi\lambda_k}{\alpha A_k} y^{2/\alpha - 1} e^{-B_k y^{2/\alpha}}$.

Assume that the channel experiences Rayleigh fading. The channel power gain $G = |h|^2$ is exponentially distributed with unit mean. And its PDF is $f_G(g) = \mathrm{e}^{-g}$.

Define $Z_k = \frac{G}{Y_k}$. Since random variables Y_k and G are independent, the PDF $f_{Z_k}(z)$ of Z_k is derived as

$$\begin{aligned} f_{Z_k}(z) &= \int_0^\infty y f_G(yz) f_{Y_k}(y)\,dy \\ &= \int_0^\infty y\,\mathrm{e}^{-yz} \frac{2\pi\lambda_k}{A_k} y^{1/\alpha} e^{-B_k y^{2/\alpha}} \frac{1}{\alpha} y^{1/\alpha - 1}\,dy \\ &= \frac{2\pi\lambda_k}{\alpha A_k} \int_0^\infty y^{2/\alpha} e^{-yz - B_k y^{2/\alpha}}\,dy. \end{aligned}$$

The achievable spectral efficiency when the user associates with the k-th tier is $\tilde{\gamma}_k = \log_2(1 + \frac{P_k Z_k}{\sigma^2})$. Its cumulative distribution function (CDF) is $\mathbb{P}\{\tilde{\gamma}_k < \xi_k\} = \mathbb{P}\{Z_k < \frac{(2^{\xi_k}-1)\sigma^2}{P_k}\}$. Thus the CDF of $\tilde{\gamma}$ is given by

$$\begin{aligned} F_\gamma(\xi) &= \mathbb{P}\{\tilde{\gamma} < \xi\} \\ &= \mathbb{P}\left\{\bigcup_k Z_k < \frac{\left(2^\xi - 1\right)\sigma^2}{P_k}\right\} \\ &= \sum_{k=1}^{2} A_k \mathbb{P}\left\{Z_k < \frac{\left(2^\xi - 1\right)\sigma^2}{P_k}\right\}. \end{aligned}$$

The differential of $F_\gamma(\xi)$ gives the PDF $f_\gamma(\xi)$ of $\tilde{\gamma}$

$$\begin{aligned} f_\gamma(\xi) &= \frac{dF_\gamma(\xi)}{d\xi} \\ &= \sum_{k=1}^{2} A_k f_{Z_k}\left(\frac{(2^\xi - 1)\sigma^2}{P_k}\right)\frac{\sigma^2}{P_k} 2^\xi \ln 2 \\ &= \sum_{k=1}^{2} \frac{2\sigma^2\pi\lambda_k 2^\xi \ln 2}{\alpha P_k} \int_0^\infty y^{2/\alpha} e^{-y\left(2^\xi - 1\right)\sigma^2/P_k - B_k y^{2/\alpha}}\,dy. \end{aligned}$$

Appendix C
Proof of Lemma 2

Since $R'_{t,\varphi}$ is deterministic, the probability $p_{t,u}^{\varphi} = \mathbb{P}\{\tilde{R}_{t,u} = r_{t,\varphi} | t, \varphi = 1, 2, \ldots, |\Phi|\}$ that user u has $\tilde{R}_{t,u} = r_{t,\varphi}$ available frequency bandwidth in time slot t is equal to the probability $\mathbb{P}\{\tilde{\varphi}_{t,u} = \varphi | t, R'_{t,\varphi} = r_{t,\varphi}\}$ that user u associates with BS φ which has $R'_{t,\varphi} = r_{t,\varphi}$ residual frequency bandwidth in given time slot t. Matrix $\mathbf{B}$ gives the prior probability $\mathbb{P}\{t | \tilde{\varphi}_{t,u} = \varphi, R'_{t,\varphi} = r_{t,\varphi}\}$. Applying Bayes formula gives the posterior probability $\mathbb{P}\{\tilde{\varphi}_{t,u} = \varphi | t, R'_{t,\varphi} = r_{t,\varphi}\} = \frac{\mathbb{P}\{t|\tilde{\varphi}_{t,u}=\varphi, R'_{t,\varphi}=r_{t,\varphi}\}p(\varphi)}{\sum_{\varphi' \in \Phi} p(\varphi')\mathbb{P}\{t|\tilde{\varphi}_{t,u}=\varphi', R'_{t,\varphi'}=r_{t,\varphi'}\}}$ which is equal to the PMF $p_{t,u}^{\varphi}$ of $\tilde{R}_{t,u}$, where $p(\varphi)$ can be obtained in matrix $\mathbf{A}$.

Appendix D
Proof of Theorem 1

The maximum achievable rate $C_{t,u}$ for user u in time slot t is $C_{t,u} = \tilde{\gamma}\tilde{R}_{t,u}$, which is a product of a continuous random variable and a discrete random variable. The CDF of the product of mixed type random variables can be calculated by

$$\begin{aligned} F_{C_{t,u}}(c) &= \mathbb{P}\{C_{t,u} < c\} \\ &= \sum_{r_{t,\varphi} > 0} \mathbb{P}\{\tilde{R}_{t,u} = r_{t,\varphi} | t, \varphi = 1, 2, \ldots, |\Phi|\} \\ &\quad \times \int_0^{\frac{c}{r_{t,\varphi}}} f_\gamma(\xi)\,d\xi \\ &\overset{(a)}{=} \sum_{r_{t,\varphi} > 0} \frac{p_{t,u}^{\varphi}}{r_{t,\varphi}} \int_0^c f_\gamma\Big(\frac{v}{r_{t,\varphi}}\Big)\,dv, \end{aligned}$$

where (a) follows by applying $\xi = \frac{v}{r_{t,\varphi}}$. Then calculating the differential of $F_{C_{t,u}}(c)$ gives the PDF $f_{C_{t,u}}(c)$ of $C_{t,u}$

$$f_{C_{t,u}}(c) = \sum_{r_{t,\varphi} > 0} \frac{p_{t,u}^{\varphi}}{r_{t,\varphi}} f_\gamma\left(\frac{c}{r_{t,\varphi}}\right). \tag{21}$$

Plugging (7) and (8) into (21) gives

$$\begin{aligned} f_{C_{t,u}}(c) = \frac{2\sigma^2\pi \ln 2}{\alpha} \int_0^\infty y^{2/\alpha} \sum_{r_{t,\varphi} > 0} \frac{2^{c/r_{t,\varphi}} p_{t,u}^{\varphi}}{r_{t,\varphi}} \sum_{k=1}^{2} \frac{\lambda_k}{P_k} \\ \times \exp\{-y(2^{c/r_{t,\varphi}} - 1)\sigma^2/P_k - B_k y^{2/\alpha}\}\,dy. \end{aligned}$$

Appendix E
Proof of Theorem 2

In problem (6), $\mathbf{J}$ and $\mathbf{T}$ are auxiliary variables to help formulate problem (6) in a standard form. So when we solve problem (6) in a RL way, $\mathbf{J}$ and $\mathbf{T}$ can be ignored.

We express the probability in $C13$ as a function of ψ

$$\begin{aligned} f(\psi) &= \mathbb{P}\{\Delta x_{t,u}^{\hat{\varphi}_{t,u}} \tilde{c}_{t,u} < b_{t,u} B\} \\ &= \int_0^\psi f_{C_{t,u}}(c)\,dc, \end{aligned}$$

where $\psi = \frac{b_{t,u}B}{\Delta x_{t,u}^{\hat{\varphi}_{t,u}}}$. $f(\psi)$ is monotone increasing with $b_{t,u}$. To minimize service delay, the agent will prompt $b'_{t,u} = 1$ for all users. So when $x_{t,u}^{\hat{\varphi}_{t,u}}$ is fixed, the optimal solution $b^*_{t,u}$ of $b_{t,u}$ is the maximum value of $b_{t,u}$ that satisfies constraint $C13$. Thus solving equation $f(\psi) = \varepsilon_1$ gives $b^*_{t,u} = \frac{\bar{\psi}_{t,u} \Delta x_{t,u}^{\hat{\varphi}_{t,u}}}{B}$, where $\int_0^{\bar{\psi}_{t,u}} f_{C_{t,u}}(c)\,dc = \varepsilon_1$ and $\bar{\psi}_{t,u}$ can be obtained from (11). In conclusion, the agent only needs to determine $\mathbf{X}$.

References

[1] *System Architecture for the 5G System*, 3GPP, TS Standard 23.501, Sep. 2018.

[2] E. Liu, X. Deng, Z. Cao, and H. Zhang, "Design and evaluation of a prediction-based dynamic edge computing system," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, 2018, pp. 1–6.

[3] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[4] C. Yao, C. Yang, and Z. Xiong, "Energy-saving predictive resource planning and allocation," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 5078–5095, Dec. 2016.

[5] J. Guo, C. Yang, and C.-L. I, "Exploiting future radio resources with end-to-end prediction by deep learning," *IEEE Access*, vol. 6, pp. 75729–75747, 2018.

[6] J. Li, X. Zhang, S. Wang, and W. Yi, "Proactive resource scheduling with time and frequency domain coordination in heterogeneous networks," in *Proc. IEEE Pers. Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, 2018, pp. 1–5.

[7] R. Atawia, H. S. Hassanein, H. Abou-zeid, and A. Noureldin, "Robust content delivery and uncertainty tracking in predictive wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2327–2339, Apr. 2017.

[8] C. Song, Z. Qu, N. Blumm, and A. L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1012, 2010.

[9] S. Sand, C. Mensing, R. Raulefs, and R. Tanbourgi, "Position-aided mobile communications," *Electron. Lett.*, vol. 46, no. 17, pp. 1232–1234, Aug. 2010.

[10] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.

[11] J. Tadrous, A. Eryilmaz, and H. E. Gamal, "Proactive resource allocation: Harnessing the diversity and multicast gains," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 4833–4854, Apr. 2013.

[12] A. M. Girgis, A. El-Keyi, M. Nafie, and R. Gohary, "Proactive location-based scheduling of delay-constrained traffic over fading channels," in *Proc. IEEE Veh. Technol. Conf. (VTC-Fall)*, Montreal, QC, Canada 2016, pp. 1–6.

[13] H. Yu, M. H. Cheung, L. Huang, and J. Huang, "Power-delay tradeoff with predictive scheduling in integrated cellular and Wi-Fi networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 735–742, Apr. 2016.

[14] J. Guo, C. Yao, and C. Yang, "Proactive resource allocation planning with three-levels of context information," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Chengdu, China, 2016, pp. 1–6.

[15] Y. Hu, S. Han, and C. Yang, "Context-aware energy saving with proactive power allocation," in *Proc. IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, Orlando, FL, USA, 2015, pp. 53–57.

[16] P. Kall and S. W. Wallace, *Stochastic Programming*. Chichester, U.K.: Wiley, 1994.

[17] R. Atawia, H. S. Hassanein, and A. Noureldin, "Robust long-term predictive adaptive video streaming under wireless network uncertainties," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1374–1388, Dec. 2018.

[18] S. Tian, X. Li, H. Ji, and H. Zhang, "Mobility prediction scheme for optimized load balance in heterogeneous networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, 2018, pp. 1–6.

[19] H. Farooq, A. Asghar, and A. Imran, "Mobility prediction-based autonomous proactive energy saving (AURORA) framework for emerging ultra-dense networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 958–971, Dec. 2018.

[20] Q. Lv, Y. Qiao, N. Ansari, J. Liu, and J. Yang, "Big data driven hidden Markov model based individual mobility prediction at points of interest," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5204–5216, Jun. 2017.

[21] D. S. Wickramasuriya, C. A. Perumalla, K. Davaslioglu, and R. D. Gitlin, "Base station prediction and proactive mobility management in virtual cells using recurrent neural networks," in *Proc. IEEE Wireless Microw. Technol. Conf. (WAMICON)*, Cocoa Beach, FL, USA, 2017, pp. 1–6.

[22] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Atlanta, GA, USA, 2017, pp. 1–9.

[23] C. Phillips, D. Sicker, and D. Grunwald, "A survey of wireless path loss prediction and coverage mapping methods," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 255–270, 1st Quart., 2013.

[24] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.

[25] W. Yi, X. Zhang, W. Wang, and J. Li, "Multi-agent deep reinforcement learning based adaptive user association in heterogeneous networks," in *Proc. Int. Conf. Commun. Netw. China (ChinaCom)*, Chengdu, China, 2019, pp. 57–67.

[26] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Nov. 2018.

[27] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, 2016, pp. 1–14.

[28] Y. Hou, L. Liu, Q. Wei, X. Xu, and C. Chen, "A novel DDPG method with prioritized experience replay," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Banff, AB, Canada, 2017, pp. 316–321.

[29] Z. Yang, K. Merrick, H. Abbass, and L. Jin, "Multi-task deep reinforcement learning for continuous action control," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Melbourne, VIC, Australia, 2017, pp. 3301–3307.

[30] C. Y. Chi, W. C. Li, and C. H. Lin, *Convex Optimization for Signal Processing and Communications: From Fundamentals to Applications*. Boca Raton, FL, USA: CRC Press, 2016.

[31] H.-S. Jo, Y. J. Sang, P. Xia, and J. G. Andrews, "Heterogeneous cellular networks with flexible cell association: A comprehensive downlink sinr analysis," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3484–3495, Oct. 2012.

[32] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional PCS networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 718–732, Oct. 2003.

Jing Li received the M.E. degree from Hebei Normal University, Shijiazhuang, China, in 2016. She is currently pursuing the Ph.D. degree with the Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include heterogeneous networks, wireless big data, and intelligent communication for 5G.

Xing Zhang is a Professor with the School of Information and Communications Engineering, Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include satellite networks, mobile edge computing, big data, and Internet of Things.

Jiaxin Zhang received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2017. From 2015 to 2016, he exchanged as a Visiting Scholar with 5G Innovation Centre, University of Surrey, U.K. Since 2017, he has been working in Beijing University of Posts and Telecommunications, and focused on the integrated satellite terrestrial networks, mobile edge computing, and green communications.

Jie Wu received the Ph.D. degree with the School of Electronic Information Engineering, Tianjin University, Tianjin, China. He joined Fujitsu Research and Development Center, Beijing, in 2015. In 2018, he joined the China Mobile Research Institute, Beijing, and currently works in research areas, such as wireless big-data-driven intelligent radio access network optimization, network resource management, and AI algorithm design.

Qi Sun received the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications. She is a Senior Researcher with China Mobile Research Institute. She has been working on the key technology and standardization of 5G radio access network. Her current research interest focuses on wireless big data and machine learning-enabled 5G and future network architecture, and protocol and algorithm design.

Yuxuan Xie received the M.S. degree in information and communication engineering from the Beijing University of Posts and Telecommunications in 2018. She joined the Green Communication Research Center, China Mobile Research Institute. Her current research interests include smart radio resource management and big data driven wireless network design.