Future Generation Computer Systems 104 (2020) 60–73

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Revised reinforcement learning based on anchor graph hashing for autonomous cell activation in cloud-RANs

Guolin Sun [a,*], Tong Zhan [a], Boateng Gordon Owusu [a], Ayepah-Mensah Daniel [a], Guisong Liu [a,b], Wei Jiang [c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[b] School of Computer Science, Zhongshan Institute, University of Electronic Science and Technology of China, Zhongshan, 528400, China
[c] German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, Germany

ARTICLE INFO

Article history:
Received 3 August 2018
Received in revised form 17 August 2019
Accepted 21 September 2019
Available online 28 September 2019

Keywords:
Reinforcement learning
Anchor graph hashing
K-means clustering
Autonomous cell activation
Cloud radio access networks

ABSTRACT

Cloud radio access networks (C-RANs) have been regarded in recent times as a promising concept in future 5G technologies where all DSP processors are moved into a central base band unit (BBU) pool in the cloud, and distributed remote radio heads (RRHs) compress and forward received radio signals from mobile users to the BBUs through radio links. In such dynamic environment, automatic decision-making approaches, such as artificial intelligence based deep reinforcement learning (DRL), become imperative in designing new solutions. In this paper, we propose a generic framework of autonomous cell activation and customized physical resource allocation schemes to balance energy consumption and QoS satisfaction in wireless networks. We formulate the cell activation problem as a Markov decision process and set up a revised reinforcement learning model based on K-means clustering and anchor-graph hashing to satisfy the QoS requirements of users and to achieve low energy consumption with the minimum number of active RRHs under varying traffic demand and user mobility. Extensive simulations are conducted to show the effectiveness of our proposed solution compared with existing schemes.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Research on the fifth-generation (5G) mobile cellular communication technology indicates that the traffic density in crowded cities or hotspot areas will reach 20 $\sim$ Tbps/km^2 in the near future. It is expected that by 2020, mobile internet will need to be delivering 1 GB of personalized data per user per day. Furthermore, traffic by 2030 is predicted to be up to 10,000 times greater than in 2010 and 100 Mbps end-user services will have to be supported [1]. To be able to support such demand, future mobile cellular networks are expected to be deployed in a very dense and multi-layered way. However, this triggers a proportional consumption on energy. From the perspective of network operators, the increasing energy costs cannot sustain future network operations. From the environmental point of view, "greenness" can be more meaningful with a comprehensive evaluation that includes both energy savings and network performance, which are the basis for energy efficiency (EE) metrics. Cloud radio access network (C-RAN) has been proposed and regarded as a promising concept in the information and communications technology (ICT) area, where base-band units (BBUs) and radios are separated [2]. Heterogeneous cloud radio access networks (H-CRANs) are considered as one of the most promising architecture to obtain a better performance in terms of efficiency, expansibility, elasticity, stability and expenditure to meet the traffic volume requirement of 5G [3]. All digital signal processing (DSP) processors are moved into a central BBU pool in the cloud, and the distributed remote radio heads (RRHs) take the responsibility of compressing and forwarding the received radio signals from mobile users to BBUs through radio links. This will reduce the overall capital expenses and operational expenses of network operators and make large-scale high-density network deployments possible. Moreover, the centralized architecture of H-CRAN makes it easy to collect and analyze statistics data of runtime system, as it motivates us to seek autonomous schemes for network energy management. H-CRANs are predicted to solve the EE problems in the future generation networks by alleviating interference and improving cooperative processing gains through cloud computing.

Recently, reinforcement learning (RL) has been advocated as a viable technology to enhance resource utilization [4]. Traditional solutions to optimize networks such as greedy linear programming and greedy search satisfy instantaneous requirements of the system whereas RL agents survey the entire network taking into account every possible state. For dynamic systems such as wireless networks where network conditions change periodically, the

* Corresponding author.
E-mail address: guolin.sun@uestc.edu.cn (G. Sun).

https://doi.org/10.1016/j.future.2019.09.044
0167-739X/© 2019 Elsevier B.V. All rights reserved.

agent selects the most appropriate policy for allocating resource in real time. In the context of H-CRAN architecture, the agent can be trained through each learning stage and then update the trained data to determine the state of each RRH in each decision epoch to implement continuous control. This paper develops a framework for EE where RL techniques are used to determine the power consumption states (sleep and active) for each RRH. The idea is to develop a revised Q-learning autonomous cell activation scheme based on anchor graph hashing and a customized physical resource allocation scheme to achieve optimal radio resource allocation for quality of service (QoS) optimization. The proposed framework can be realized in two steps: Firstly, we decide the states of RRHs as active or inactive using a revised reinforcement learning algorithm. Secondly, we set up a flexible resource allocation module based on the active RRH set to maximize the aggregated QoS satisfaction of all user equipment (UEs).

Unlike most of the previous model-driven works for energy optimization just presented algorithms optimizing a certain objective for the current timeslot (or time frame) [2,5], our proposed revised Q-learning-based framework makes a sequence of cell activation decisions to minimize total energy consumption while satisfying QoS demand of UEs for the whole operational period, which is characterized by model-free or data-driven methods using hash codes generated from anchor graph hashing as inputs. The main difference between the traditional Q-learning, deep reinforcement learning and our proposed revised Q-learning technique is that, the traditional Q-learning technique used in [6] takes vectors as inputs which mostly do not consider user association and mobility patterns of users. Deep reinforcement learning-based scheme in [7,8] deals with online training and has a slow convergence rate. Our revised Q-learning technique uses *relational matrix* from the network as the input, which is a UE-by-RRH matrix, referred to in Definition 1. Since a matrix cannot be used as input for the Q-learning agent, we convert the relational matrix into a hash code. Our aim is to map the input space to similar codes in the Hamming space. The distance among the hash codes are also called the hamming distance. The state space of each RRH comprises the on/off state and the traffic load on the RRH. This makes the combinatorial state space continuous and infinite with lack of immediate reward signal. With this problem, RL technique specifically, Q-learning can be applied to achieve a solution, but it can only deal with limited state space. We can decompose or discretize the state space into simpler state spaces and learn them independently. In this way, learning is accelerated since every separate task is easier to learn. If the discretization is too fine, curse of dimensionality occurs and the state space increases exponentially. To solve the problem of curse of dimensionality, we introduce k-means clustering which reduces the state space size by dynamic abstraction where the learning agent gathers knowledge of the environment and divides it into sub-tasks of different clusters [9]. However, the sub-tasks may change throughout the learning process. K-means clustering is used to form clusters based on a rough estimate of the environment. Based on the rough estimate, the agent can explore the environment to improve on its decisions and anchor graph hashing maps the clusters to hash codes. To efficiently solve this problem, we first use a revised Q-learning method to solve the cell activation problem and formulate the resource allocation problem for users as a convex optimization problem. Our motivation is to achieve the balance between EE and QoS to satisfy infrastructure providers (InPs) and mobile users via resource allocation customized for rate, delay and jitter optimization and decoupled from cell activation techniques in an H-CRAN system. The main contributions of this paper can be summarized as:

- We propose an autonomous energy management framework using cell activation techniques for a customized network. We design a revised Q-learning model which takes hash codes as input states with a reduced size of state space considering varying resource demand, traffic profile and user mobility.
- In this framework, we formulate the EE-QoS optimization as two models; resource allocation for rate-constrained users and resource allocation for delay-constrained users, but unified with aggregated QoS satisfaction.
- Comprehensive simulations are conducted to verify the significance of the proposed work. It is demonstrated that the QoS satisfaction of users are greatly improved and energy consumption is minimized with changing user mobility under the proposed scheme.

The remainder of this paper is organized as follows. In Section 2, we present related works. Section 3 presents the system model in terms of network model, traffic model, energy model and utility model. Section 4 provides the problem formulation and our proposed RL-based autonomous energy management framework. Simulation results and analysis are discussed in Section 5. We conclude this work in Section 6.

2. Related work

The EE and QoS performance metrics have become design goals as the discussion on energy consumption continues to grow across every field. It has become a requirement for network engineers and scientists to develop systems that manage energy efficiently. Authors in [10] studied energy efficient wireless communications and identified energy-efficient resource allocation as one of the key challenges of 5G technology. In C-RAN, baseband and processing functionality of a network are virtualized and shared among physical units. This architecture improves EE in the sense that the RRHs have fewer functions. In [11], the authors jointly considered RRH selection and power minimization as the resource allocation problem in group sparse beam forming for green C-RAN. The authors extended their work to reduce the computational complexity in selecting RRH using Lagrangian dual methods in [12]. In [13], the effect of optimizing data-sharing and the compression on EE were studied in C-RAN. By minimizing the total power consumption in the network, they proved that higher EE was dependent on the user target transmission rate. In [2], *Luo et al.* proposed a joint downlink and uplink mobile users access point association and beam-forming design for interference management and energy consumption minimization in C-RAN. Authors in [14] also proposed an enhanced soft fractional frequency reuse scheme. In this scheme, they formulated a joint optimization problem with resource block assignment and power allocation for interference mitigation in order to maximize EE performance in H-CRANs.

A convex optimization-based framework was proposed in [5] for energy-efficient global radio resource management in heterogeneous wireless networks. With stochastic arrivals of known rates intended for users, the smallest set of BSs is activated with jointly optimized user association and spectrum allocation to stabilize the network first and then minimize the delay. Joint optimization by cell activation or cell coverage adjustment, user association, and sub-carrier allocation has been investigated in [15,16]. This was done under the constraints of maintaining an average sum rate and rate fairness. The authors argued that energy consumption is dependent on both the spatial–temporal variations of traffic demands and the internal hardware components of sc-BSs. Several studies in recent times also suggested a scheme, as known as multiple base station scheduling (MBSS) in [17]. Due to the computational complexity of MBSS, authors in [18]

proposed a low complexity flexible flow scheduling algorithm to compensate for the energy consumption introduced by the increasing dimension of ultra-dense nodes. Trade-offs between QoS and EE for users with different traffics was presented in [19]. In [20], the authors studied the user association problem aiming at maximizing the EE of the network for the downlink of heterogeneous networks (HetNets). The goal of minimizing the system energy consumption and also maximizing the ratio of the peak-signal-to-noise-ratio was considered in [21] but only for QoE-aware energy efficiency and QoE-aware spectral efficiency.

Reinforcement learning can be widely utilized in many applications with different optimization objectives. Some authors investigated the opportunities, challenges and potential applications of artificial neural networks (ANN) in future wireless networks in [22]. They presented some key types of neural networks and various wireless communication problem that can be solved using ANNs. A framework for offloading the backhaul and fronthaul loads in C-RAN was proposed in [23]. The authors jointly combined machine learning tools of echo state networks (ESNs) with a sublinear caching approach to develop a novel algorithm which enables BBUs to predict content request distribution of users with limited information on the network states and also helps the BBU to compute content request percentage using a few samples. Authors discussed the problem of resource management in wireless virtual reality in [24]. They formulated a non-cooperative game between sc-BSs and a distributed algorithm based on machine learning tools in ESNs to propose a solution to the game. *Xu et al.* proposed a framework which uses reinforcement learning to achieve an optimal solution for power-efficient resource allocation for beam forming problem in [8].

To the best of our knowledge, there are lack of solutions to capture dynamic topology, traffic distribution and mobility information of users with model-free RL techniques. In this paper, by the abstraction of UE-RRH association matrix as anchor graph and by capturing dynamic traffic load, network topology and user mobility, the power manager adopts a model-free RL technique to adaptively determine the suitable action for turning on/off of the RRHs reducing the power consumption, and simultaneously improving QoS satisfaction.

3. System model

A. System architecture

The system of autonomous cell activation and customized physical resource allocation for energy consumption and QoS optimization is made up of a macro base station (MBS), a group of BBUs, several RRHs and UEs, as shown in Fig. 1. The RRHs underlay the MBS in a same spectrum resource pool and all digital signal processing (DSP) processors are moved into a centralized BBU pool in the cloud. The interference between MBS and RRHs can be suppressed by advanced multiple-input and multiple-output (MIMO) and interference mitigation techniques. The MBS is linked to the BBU pool by the backhaul interface for control exchange whiles the RRHs are connected by the fronthaul interface for compressing and forwarding the received radio signals from mobile users to BBUs forming a coordinated multipoint processing (CoMP). This greatly increases the capacity and time delay constraints on the fronthaul links of the network efficiently by having the MBS and RRHs cooperate in their transmissions. Distributed RL agents are deployed on each BBU pool, which control the cell activation dynamically and the UEs are connected to the RRHs. The UEs and RRHs report their state information (SI) to the RL agent in the BBU pool. The SI includes UE-RRH association, received signal strength, achieved data rate, delay, traffic arriving rate of the UEs and traffic load on each RRH. The traffic arriving rate of each UE, which depends on the type of application the UE is subscribed to, is available to the BBU cloud. In a case where this information is not accessible, the service provider can report such kind of information to the BBU cloud. The cell activation process is performed by the RL agent by way of sending an on or off switching decision to the RRHs. The RL agent monitors the mobility of the UEs and how they affect its decisions. The reward that is calculated by the RL agent combining user satisfaction and power consumption of all RRHs in a centralized manner, is stored in each BBU cloud and made available to every RRH in the cloud.

The proposed framework of energy optimization scheme has three hierarchies. Firstly, user association between UEs and RRHs is established via user admission control. Then, the RL agent executes cell activation using the revised Q-learning technique to select the active RRH set. The RL agent dynamically monitors changes in user population, user association, traffic load distribution, QoS demand and energy cost caused by the dynamics of the mobile environment. Lastly, the physical resource allocation module tries to satisfy the QoS requirement of UEs with the specific action from the RL agent, which results in the active RRH set. The result of resource allocation serves as the reward which is fed back into the revised Q-learning-based cell activation module. Once convergence is achieved, the RL agent autonomously outputs the best decision on actions to the environment. For simplicity, we summarize all the notations used in this paper as listed in Table 1.

B. Network model

Let $j \in \mathcal{J} = \{1, 2, \ldots, |J|\}$ be a set of RRHs. For each RRH $j \in \mathcal{J}$, a set of UEs $i \in \mathfrak{T} = \{1, 2 \ldots, |I|\}$ are connected to it. Each UE is subscribed to a specific application $k \in \mathcal{K} = \{1, 2 \ldots, |K|\}$ which has a unique traffic arriving rate λ_{ij}^{k}. The system bandwidth for RRH j is denoted by B Hz and its total transmission power consumption is P_j^t watts. In the system model, the path-loss is calculated as follows:

$$PL(d_{ij}) = 20 * log\,(F) + 20 * log\left(d_{ij}\right) + 32.4, \tag{1}$$

where F is the frequency band and d_{ij} is the distance between UE i and RRH j. We consider the channel model [11] as;

$$g_{ij} = 10^{-PL(d_{ij})/20}\sqrt{b_{ij}}\zeta, \tag{2}$$

where $PL(d_{ij})$ is the path loss, b_{ij} is the antenna gain and ζ is the shadowing small-scale fading.

The shadowing small-scale fading ζ is assumed as a Gaussian random variable with zero mean and standard deviation δ equal to 8 dB [25]. One major use case of H-CRAN is the ability to coordinate transmission among RRHs forming a CoMP. Therefore, we classify the UEs into two groups, normal users and edge users. The edge users are defined as the 10% of all UEs, who have the lowest signal-to-interference-plus-noise-ratio ($sinr$) and will be served in CoMP transmission mode. We assume that, a normal user is associated with only one RRH for transmission whiles an edge user is associated with more than one RRH for transmission via beamforming. The $sinr$ experienced by normal UE i associated with RRH j is given by [13],

$$sinr_{ij} = \frac{\left|g_{ij}^{H} w_{ij}\right|^2}{\sum_{u \neq i}\left|g_{ij}^{H} w_{uj}\right|^2 + \sigma^2}, i \in I, \tag{3}$$

where g_{ij} is the channel gain from RRH j to UE i, w_{ij} is the beamforming weight from RRH j to UE i and σ^2 is the power spectral density of additive white Gaussian noise. With (3), the achieved data rate r_{ij} of UE i connected to RRH j is given as,

$$r_{ij} = B\log_2(1 + sinr_{ij}), \tag{4}$$

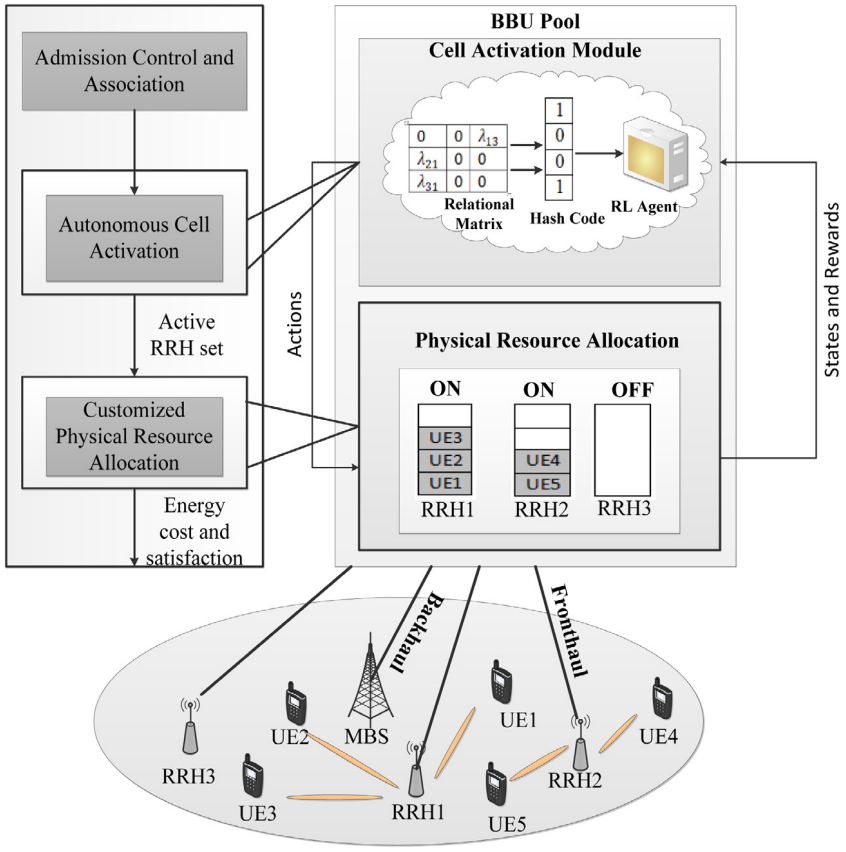


Fig. 1. Scenario and system architecture.

Table 1
Table of notations.

System parameters		Algorithm parameters	
Symbol	Meaning	Symbol	Meaning
$\mathcal{J}$	A set of RRHs	M	MDP Tuple: State Space S, Action Space A and Reward $R(s, a)$
$\mathfrak{T}$	A set of mobile UEs	t	Stage number
$\mathcal{K}$	Set of application subscribed to by UE	$U_\pi(s)$	State-value function for a policy π
B	System bandwidth	γ	Discount factor
F	Frequency band	θ	Positive parameter: temperature
d_{ij}	Distance between UE i and RRH j	Y	Relational matrix
ζ	Shadowing small scale fading	$\tilde{S}$	Clustered state space
δ	Standard deviation	$\overline{S}$	Continuous state space
$SINR$	SINR between UE and RRH	m	Number of cluster centroid
g_{ij}	Channel gain of UE i and RRH j	C	True adjacency matrix
σ	Power spectral density of additive white Gaussian noise	Z	Truncated adjacency matrix
x_{ij}	Binary indicator for UE-RRH association	$\hat{C}$	Approximated true adjacency matrix
r_{ij}	Achieved data rate of UE i on RRH j	V	Spectral embedding matrix
λ_{ij}^k	Traffic arriving rate of UE i on RRH j based on application k	Φ	Column-orthonormal matrix
τ_{ij}	Average delay of UE i associated with RRH j	$H(\cdot)$	Hash function
T	24-h scan period	P_j^t	Total Power consumption of RRH j
ρ_j	Traffic load on RRH j	ξ	Satisfaction variable
P_j^c	Constant power of RRH j	Ł	Mean packet size (in bits)
P_j^l	Load-aware transmit power of RRH j	E	Number of Active RRH

where B is the channel bandwidth. The traffic arriving rate of UE i subscribed to application k on RRH j is denoted by λ_{ij}^k, which is dependent on the application type. The traffic load ρ_j on RRH j is defined as the sum of the achieved data rates of the UEs connected to the RRH, which can be expressed as $\rho_j = \sum_i \sum_k x_{ij} r_{ij}^k$, where x_{ij} denotes an association indicator between UE i and RRH j. If $x_{ij} = 0$, there is no association between the UE and RRH; otherwise, $x_{ij} = 1$. The service time of QoS traffic for UE i associated with RRH j is $t_{ij} = \frac{Ł_i}{r_{ij}} = \frac{1}{r_{ij}^*}$, where r_{ij}^* denotes the normalized achievable rate with respect to the average packet length. Note that the packet length for UE i is independent and exponentially distributed with the mean packet size (Ł)) in bits. Based on M/M/1 queuing theory, the average delay experienced by UE i on RRH j is

$$\tau_{ij} = \frac{1}{\frac{1}{t_{ij}} - x_{ij}\lambda_{ij}^k} = \frac{1}{r_{ij}^* - x_{ij}\lambda_{ij}^k}. \tag{5}$$

C. Traffic model

The system load is dependent on the user traffic arriving rate and the on-off mode of RRHs. Due to the dynamics in user behavior on traffic demand, analytical model is intractable. We therefore adopt a data-driven model from the EU FP EARTH project [26]. In our scenario, we monitor the spatial–temporal traffic distribution in the network over a 24-h period. We assume that the period is long enough so that the traffic load profile in one period repeats in the next period. In mobile environment, the type of active UEs, their mobility pattern, user population, distribution and environment noise vary over this period. Without loss of generality, an ideal traffic profile is adopted based on the daily trapezoidal traffic pattern [18]. Given a time $t = 1, 2, 3, \ldots, T$, the traffic model is defined with the angular coefficient v as

$$f(t) = \begin{cases} 1 - vt; & 0 \le t \le \frac{1}{v} \\ 0; & \frac{1}{v} \le t \le T - \frac{1}{v} \\ 1 + v(t - T); & T - \frac{1}{v} \le t \le T, \end{cases} \tag{6}$$

where T represents a 24-h scan period, v represents the slope and $f(t)$ is a normalized value between 0 and 1. If v is equal to 1/10, then we move the $f(t)$ to $f(t) + 4$, which is close to the on-site measurements in [26].

D. Energy model

To make a correct estimation of the RRH energy consumption, we need to consider two types of power consumption, i.e. constant power, which is generated by the always-on components of the RRHs, e.g. transceivers and power supply, and power due to the traffic load which comes from the power amplifier [27]. We define the total energy consumption for each RRH at time t as follows:

$$P_j^t = P_j^c + \rho_j^* . P_j^l, \tag{7}$$

where P_j^c denotes the constant power consumption for active RRHs, P_j^l denotes the load-aware transmit power of RRH j for transmission which depends on the load and ρ_j^* is the normalized traffic load on RRH j at time t. P_j^s is the constant power consumption for sleeping RRHs.

In the H-CRAN architecture, inactive RRHs are put to sleep in order to conserve energy. Our proposed revised Q-learning based cell activation scheme provides flexibility for managing energy consumption. The H-CRAN control unit dynamically optimizes the total expected and cumulative energy during the entire operational period instead of the instantaneous energy consumption in a decision period.

E. Utility model

Based on the objective of the proposed scheme, we can know the precondition of minimizing energy consumption of the C-RAN is to ensure that we satisfy the QoS requirement of UEs. To ensure UE satisfaction, the required transmission rate and delay should be guaranteed. Due to the dynamics in UE behavior, traffic demand per connection and service elasticity, we model the utility with a sigmoid function. The utility function maps the perceived achievable rate with the level of UE satisfaction [6]. The satisfaction of UE i on rate is,

$$\xi(r_i) = \frac{1}{1 + e^{-\eta(r_i - r_i^{min})}}, \tag{8}$$

where r_i^{min} is the minimum rate requirements of UE i and η is a constant which determines the shape of the satisfactory curve. In addition, r_i is the transmission rate for UE i. It is easy to verify that: (1) $\xi(r_i)$ is a monotonic increasing function with respect to r_i, because individual users will feel more satisfied if they receive higher throughput above their minimum demand and vice versa; (2) $\xi(r_i)$ of each UE i is scaled between 0 and 1, i.e. $\xi(r_i) \in [0, 1]$. The satisfaction on delay is given mathematically as;

$$\xi(\tau_i) = \frac{1}{1 + e^{-\eta(\tau_i^{max} - \tau_i)}}, \tag{9}$$

where τ_i^{max} is the maximum tolerant delay, which is required to satisfy the upper bound delay for UE i.

4. Problem formulation

In this section, we formulate the energy optimization problem in H-CRAN with a revised Q-learning agent for autonomous cell activation, illustrated in Fig. 2. Most of the existing works take one-dimensional vectors as states for the Q-learning agent, which include very little information about network dynamics. In order to capture spatial–temporal dynamics in a mobile environment, we use a graph, defined as a relational matrix, which has more information such as the association between UEs and RRHs, the arriving rates of traffic, the transmission rates of UEs and UE satisfaction. However, the state space size is very large, which makes it difficult for the Q-table to converge. Anchor graph clustering is used to reduce the state space size and anchor graph hashing is employed to map the graphs to hash codes. The discrete characteristics of hash code match the RL agent very well. With the output action of RL agents, customized physical resource allocation is done to generate the report of UE satisfaction and energy calculation, which are the rewards that are fed back to the RL agents.

A. Offline anchor graph clustering

In our scenario, we define the network image by a relational matrix Y between UEs and RRHs, which can include the association relationship between the UEs and RRHs in the network as well as the subscription relationship, UE traffic arriving rate, UE transmission rate and even delay, which reflect the dynamics caused by UE mobility and noise in the radio environment.

Definition 1 (*Relational Matrix*)**.** Mathematically, we define relationships between all $|I|$ UEs and $|J|$ RRHs as a matrix $Y \in \mathrm{R}^{|I| \times |J|}$ as;

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1j} \\ \vdots & \ddots & \vdots \\ y_{i1} & \cdots & y_{|I||J|} \end{bmatrix}$$

The entries y_{ij} could be the arriving rate of traffic λ_{ij}^k, UE transmission rate r_{ij}, satisfaction $\xi(\cdot)$ or the transmission rate to traffic arriving rate ratio r_{ij}/λ_{ij}^k. Let an entry be represented as

$$y_{ij} = \begin{cases} \lambda_{ij}^k, & if\ x_{ij} = 1 \\ 0, & otherwise, \end{cases} \tag{10}$$

where x_{ij} is a binary association indicator between a RRH and a UE, in that, $x_{ij} = 1$ means there is a connection between UE i and RRH j and $x_{ij} = 0$ means otherwise. The λ_{ij}^k is the traffic arriving rate between UE i, who is assumed to subscribe to the application k and RRH j. Therefore, the sum of all the traffic load of UEs on RRH j can be denoted by ρ_j.

The relational matrix is a sparse matrix of which each column contains only one nonzero entry, except for at most 10% of UEs working in multipoint joint transmission mode. Then, we flatten the relational matrix Y to a one-row vector $\bar{s}$ as a single sample. We then stack all of the individual one-row vectors together to obtain a training set matrix $\bar{S} \in R^{|U| \times |I| \times |J|}$, where $|U|$ is the number of training set samples and $|I| \times |J|$ is the number of elements in one-row vector $\bar{s}$. Since the original volume of raw topology samples is very large, we introduce an offline anchor graph clustering method to compress the sample size before hash mapping. Anchor graph is a kind of graph that basically defines the similarities between two vertices: *data points* and *anchors* [28]. An anchor graph uses a small set of m points called anchors to approximate the data structure of neighboring nodes

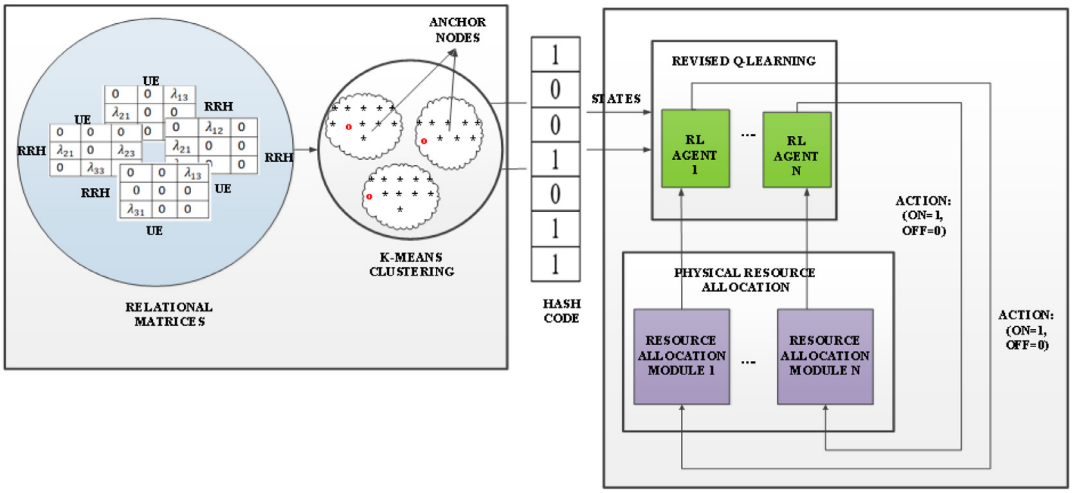


Fig. 2. The revised reinforcement-learning framework.

or sample points u. An anchor graph is normally undirected, that is, there is no distinction between the two vertices associated with each edge.

(i) State space discretization: We discretize the state space based on the centroids and then partition the centroids based on *K-means* algorithm. With clustered centroids as anchor nodes, we partition $|U|$ sample points into $|M|$ clusters.

Definition 2. Given $|M|$ centroids in a continuous state space as $\tilde{S} = \{\tilde{s}_m | m = 1, 2, \ldots, |M|\}$, the state space can be partitioned into $|M|$ clusters. The partitions of state space is $\bar{S} = \{\bar{s}_u | \bar{s}_u \in \mathbb{R}^d,\ u = 1, 2, \ldots, |U|\}$, where the continuous state space region for a cluster centroid $\tilde{s}_m$ is expressed as

$$\bar{S} = \left\{ \bar{s}_u \in \mathbb{R}^d | m = min \left\{ arg \min_m \|\bar{s}_u - \tilde{s}_m\|, \right\} \right.$$
$$\left. m = 1, 2, \ldots, |M| \right\}. \quad (11)$$

$\bar{S}$ denotes the original continuous state space, $\tilde{S}$ is the discrete state space, $\bar{s}_u$ is a specific state of all training samples in $\bar{S}$ and $\tilde{s}_m$ represents is a specific state of anchor nodes in $\tilde{S}$. Each anchor node has a length of $|I| \times |J|$.

(ii) K-means based centroid partitioning: Given a set of cluster centroids, we can partition the continuous state space by transforming it to obtain a suitable distribution of cluster centroids. We predefine the number of maximal cluster centroids $|M|$, the distance threshold q_d, the number of training set samples u and obtain the continuous state $\bar{s}_u$. Then, we compute the distances between the continuous state $\bar{s}_u$ and each cluster centroid. If the distance between the continuous state $\bar{s}_u$ and a cluster centroid $\tilde{s}_m$ exceeds the distance threshold q_d, and the number of current cluster centroids is less than the maximal cluster centroid $|M|$, we add a new cluster centroid $\tilde{s}_{m+1}$ to the location of the continuous state. Otherwise, we adjust the location of the cluster centroids using K-means clustering algorithm and update the process until $u = |U|$. Lastly, we approximate the adjacency matrix C using Eq. (15) and perform offline hash mapping on each centroid using Eq. (16). This offline *K-means* clustering is performed on the $|U|$ training samples to obtain $|M|(|M| \ll |U|)$ cluster centroids with m as anchors as;

$$\tilde{S} = \left\{ \tilde{s}_m \in \mathbb{R}^{(|I| \times |J|)} \right\}_{m=1}^{|M|}. \quad (12)$$

This makes clustering very fast, thus speeding up training significantly.

B. Anchor graph hashing

After creating clusters with identified anchor graphs, we map the clusters to hash codes by anchor graph hashing.

Definition 3 (*True Adjacency Matrix*)**.** Let the true adjacency matrix be represented by the matrix C as

$$C = \left\{ c_{ij} \right\} \in [0, 1]^{|U| \times |U|}, \quad (13)$$

where c_{ij} is the similarity between u_i and u_j, $i, j \in [1, |U|]$. An adjacency matrix is a matrix such that its elements are values other than zero when there is an edge between two vertices, and zero when there is no edge. The true adjacency matrix C is approximated to $\hat{C}$ to reduce the computational complexity using Eq. (15) by applying *K-means* clustering on the training samples $|U|$. The truncated adjacency matrix Z results in a graph Laplacian L. The eigenvectors of $\hat{C}$ is easily solved by utilizing its low-rank property resulting in eigenvectors–eigenvalues pairs.

Definition 4 (*Truncated Adjacency Matrix*)**.** Anchor Graph defines the *truncated adjacency matrix* Z with similarities of all the $|U|$ training *sample points* measured with respect to the m *anchors* as,

$$Z_{um} = \begin{cases} \frac{\exp(-\frac{D^2(\bar{s}_u, \tilde{s}_m)}{w})}{\sum_{m' \in <u>} \exp(-\frac{D^2(\bar{s}_u, \tilde{s}_{m'})}{w})}, & \forall m \in <u> \\ 0, & otherwise, \end{cases} \quad (14)$$

where $\langle u \rangle \subset [1; |M|]$ denotes the indices of m' nearest anchors of sample $\bar{s}_u$ in $\tilde{S}$ according to a distance function $D(\cdot)$, w is the bandwidth parameter and $z_{um} = \exp(-\frac{D^2(\bar{s}_u, \tilde{s}_m)}{w})$.

With Definitions 3 and 4, anchor graph provides a powerful approximation of the true adjacency matrix C to [28];

$$\hat{C} = Z \Lambda^{-1} Z^T, \quad (15)$$

where $\Lambda = diag(Z^T 1 \in \mathbb{R}^{m \times m})$ with $1 = [1, 1, \ldots, 1]^\top \in \mathbb{R}^m$

Theorem 1 (*Offline Hash Mapping*)**.** *Given u training samples and truncated adjacency matrix Z, binary hash codes are mapped from the spectral embedding matrix V, which is calculated as follows;*

$$V = ZN. \quad (16)$$

Proof. The resulting graph Laplacian of the anchor graph is given by; $L = I - \hat{C}$, where I is an identity matrix. We solve the eigenvectors of $\hat{C}$ by utilizing its low-rank property as $\hat{C} = Z \Lambda^{-1} Z^T = (Z \Lambda^{-1/2})(Z \Lambda^{-1/2})^T$. We solve the eigenvalue system of a small

$m \times m$ matrix $\hat{C}_m = (Z\Lambda^{-1/2})^T(Z\Lambda^{-1/2}) = \Lambda^{-1/2}Z^TZ\Lambda^{-1/2}$, resulting in $r(< m)$ (r largest eigenvalues) eigenvector–eigenvalue pairs $\{(\alpha_k, \beta_k)\}_{k=1}^{r}$ where $1 > \beta_1 > \cdots > \beta_r > 0$. Obviously, $rank\left(\hat{C}_m\right) = rank\left(\hat{C}\right) = r$. Lastly, we obtain the desired spectral embedding matrix V as;

$$V = \sqrt{u}Z\Lambda^{-1/2}\varphi\Sigma^{-1/2} = ZN,$$

where $N = \sqrt{u}\Lambda^{-1/2}\varphi\Sigma^{-1/2} = [n_1, \ldots, n_r]^{m\times r}$, $\varphi = [\varphi_1, \varphi_2, \ldots., \varphi_r] \in \mathbb{R}^{m\times r}$ is column-orthonormal, $\Sigma = diag\,(\beta_1, \ldots, \beta_r) \in \mathbb{R}^{m\times r}$ and $n_m = \sqrt{u/\beta_m}\Lambda^{-1/2}\varphi_m$. Therefore, we propose an offline training algorithm for anchor graph clustering and hashing based on the above, and make a summary in Algorithm 1.

On-line hash function mapping

Eq. (16) generates hash codes only for the available points during training. Therefore, a general hash function is needed in order to predict the hash code of any incoming sample point $\bar{s}$. We generalize the eigenvectors of the anchor graph Laplacian to the eigenfunctions $\left\{\varnothing_k: \mathbb{R}^d \to \mathbb{R}\right\}_{k=1}^{r}$ such that the hash functions can be defined as $H_k(\bar{s}) = sgn\,(\varnothing_k(\bar{s}))\,(k = 1, \ldots, r)$, where $\varnothing_k(\bar{s}) = n_k^T z(\bar{s})$ and $z(\bar{s}) = \frac{\mu_{1z_{um}}, \ldots. \mu_{mz_{um}}}{\sum_{m=1}^{|M|}\mu_{mz_{um}}}$. We then create the "out-of-sample" extension V to their corresponding eigen-functions using the Nyström method in [29]. The online hashing will be included in the revised Q-learning.

C. Revised Q-learning

Reinforcement learning is a form of machine learning technique whereby an agent interacts solely with an environment, without requiring additional information about the environment except for awareness of the environment states, possible (enabled) actions from its current state, and the obtained rewards after performing a specific action [30]. There are a number of reinforcement learning technique variations such as Q-learning, deep Q-learning and double Q-learning. Q-learning is a model-free reinforcement learning algorithm for discrete state spaces. Because of the discrete characteristics of state feature extraction with hash codes in our scenario, Q-learning is adopted. Firstly, we begin to model the cell activation problem in H-CRAN as a Markov decision process (MDP).

(i) Markov decision process: The interaction between the Q-learning agent and the radio access network environment can be represented as a tuple, $\mathcal{M} = \left[S, A, R\,(s, a)\,, S'\right]$, where S represents the set of possible states, A is the set of actions, $R(s, a)$ represents the reward achieved when an action a, in state s, is selected and s' is the next state. At any stage t with a traffic load state $s^{(t)}$, the RL agent chooses an action $a^{(t)}$ to either turn on or off an RRH. Let $U\left(\left[s^{(0)}, s^{(1)}, \ldots..., s^{(t)}\right], a^{(t)}\right)$ represent a Markov chain utility. Since future rewards are unpredictable, the long-term reward of state $s^{(t)}$ at stage t is the sum of discounted rewards which is given by,

$$R(s^{(t)}) + \sum \gamma^t R\left(s^{(t)}\right) + \sum \gamma^{t+1} R\left(s^{(t)}\right) + \cdots, \tag{17}$$

where γ is the discount factor ranging from 0 to 1. If $\gamma = 0$, it means we only care about the current reward. $0 < \gamma \leq 1$ indicates that we care about future rewards. We denote the state-value function of an arbitrary policy at the stage t as;

$$U_\pi\,(s) = \mathbb{E}\left\{\sum_{t=0}^{\infty}\gamma^t R(s^{(t)})\right\}. \tag{18}$$

The resource image state $s^{(t)}$ transforms into $s^{(t+1)}$ at stage $(t+1)$ with a transition probability as;

$$P\left(s'|s^{(t)}, a^{(t)}\right) = \begin{cases} 1, s' = s^{(t+1)} \\ 0, otherwise. \end{cases} \tag{19}$$

The goal of MDP is to find an optimal policy π^* to maximize future reward of the decision agent. A policy is a mapping from states to actions. From Markov property, the policy π can be further expressed as;

$$U_\pi\,(s) = \mathbb{E}\left\{R(s^{(t)}, a^{(t)}) + \sum_{s'} P(s'|s^{(t)}, a^{(t)})U_\pi(s')\right\}, \tag{20}$$

where $U_\pi(s')$ is the expected utility given the optimal policy. The state-value function for optimal policy based on the Bellman's equation [30] is given as;

$$U_{\pi^*}\,(s) = \arg\max_{a^{(t)}\in A}\left\{R(s^{(t)}, a^{(t)}) + \gamma\sum_{s'} P(s'|s^{(t)}, a^{(t)})U_\pi(s')\right\}, \tag{21}$$

where $R(s^{(t)}, a^{(t)})$ is the present reward, γ is the discount factor, $U_\pi\,(s)$ is the present utility and $U_\pi(s')$ is the future utility.

(ii) Revised Q-learning framework: The offline anchor graph clustering generates cluster centroids i.e. anchor graphs with each centroid having a unique hash code. When a new sample state arrives, it is compared with a general hash function $H_k(\bar{s})$ and the closest hash code is selected for the new state in an online manner. We formulate the interface between RL agents and environment by defining states, actions, rewards and next state.

State(s): As mentioned above, the purpose of Q-learning is to minimize the number of active RRHs while satisfying the QoS requirements of UEs. The components of the state space are obtained from the anchor graph hashing discussed in Theorem 1. The resource images generated from anchor graph hashing are converted into hash codes which are fed into the RL agent in the BBU cloud as input states. The state space for offline training is obtained from the desired spectral embedding matrix V in Eq. (16) which has m states and r bits for each hash code. The size of the Q-table is dependent on m i.e. an increase in m increases the state space size. For online Q-learning, we resort to a general hash function $H_k(\bar{s})$ to cluster a new state sample $\bar{s}$ to the closest anchor graph hash code $H_k(\bar{s}^*)$.

Action(a): The action to be performed is the switching decision that is made by the agent on the RRHs. Each RRH corresponds to two actions, switching on or off. For any RRH j, the actions can be represented as $a_j \in 0, 1$, $a_j = 0$ indicates switching off RRH j to turn it to sleep, and $a_j = 1$ indicates switching on RRH j to turn it on. We assume that the system begins at stage t with a resource image state of $s^{(t)}$. The agent can select an action based on two objectives: exploration and exploitation. The learning agent chooses an action $a^{(t)}$ in the state $s^{(t)}$ of the stage t with a Boltzmann distribution probability [31]

$$P^{(t)}\left(s^{(t)}, a^{(t)}\right) = \frac{exp\left\{p(s^{(t)}, a^{(t)})/\theta\right\}}{\sum_{a'\in A} exp\left\{p(s^{(t)}, a')/\theta\right\}}, \tag{22}$$

where θ represents a positive parameter called temperature and $p(s^{(t)}, a^{(t)})$ denotes the possibility that the agent selects action $a^{(t)}$ at state $s^{(t)}$. The action chosen is updated after every stage.

Reward(R): Reward is the feedback received from the environment after performing an action in a specific state. Therefore, the reward needs to reflect the purpose of the Q-learning algorithm. In our case, the reward is a function of average QoS satisfaction of the UEs and energy consumption of the system. Since the optimal strategy of Q-learning is to find the action with the maximum Q-value in the Q-table for each state, we define the reward as follows:

$$R\left(s^{(t)}, a^{(t)}\right) = \frac{1}{E} + \omega * \xi\,(\cdot)\,, \tag{23}$$

where E is the number of active RRHs, $\xi\,(\cdot)\,\epsilon[0, 1]$ is an indicator for the QoS satisfaction of UEs, with utility function defined in

Algorithm 1: K-means based centroid partitioning and offline hash mapping

1: Initialization $|M|$, q_d, $|U|$;
2: **while** $u < |U|$ **do**
3: Obtain continuous state $\bar{s}_u$
4: Compute the distance $D(\cdot)$ between $\bar{s}_u$ and each centroid $\tilde{s}_m$
5: Obtain Z_{um} using Eqn. (14)
6: **if** $\left|\left|\bar{s}_u - \tilde{s}_m\right|\right| > q_d$ and $m > |M|$ **then**
7: Add a cluster centroid $\tilde{s}_{m+1}$ at the location of $\bar{s}_u$
8: **else**
9: $\tilde{s}_m$ is the cluster centroid of $\bar{s}_u$ from Eqn. (12)
10: Adjust the location of $\tilde{s}_m$ and let $u = u + 1$
11: **end if**
12: **end while**
13: Approximate adjacency matrix C with Z using Eqn. (15)
14: Perform offline hash mapping on each centroid $\tilde{s}_m$ to obtain V in Eqn. (16)

(8) or (9) and $\omega > 0$. The optimal strategy, denoted by π^* is used to maximize the action-value function of each state, s. The optimality equation in terms of Q only can be expressed with Eq. (21) as;

$$Q\left(s^{(t)}, a^{(t)}\right) = R(s^{(t)}, a^{(t)}) + \gamma \sum_{s'} P(s'|s^{(t)}, a^{(t)}) \max_{a^{(t)} \in A} Q(s^{(t+1)}, a^{(t+1)}) \forall s, \pi. \tag{24}$$

The Q-learning agent learns the optimal Q-values in an iterative manner based on information available in the BBU cloud. The value iteration Q-learning process can be expressed with Eq. (21) as;

$$Q^{(t+1)}\left(s^{(t)}, a^{(t)}\right) \leftarrow R(s^{(t)}, a^{(t)}) + \gamma \sum_{s'} P(s'|s^{(t)}, a^{(t)}) \times \max_{a^{(t)} \in A} Q^{(t)}(s^{(t+1)}, a^{(t+1)}). \tag{25}$$

By iterating and updating the $Q(s^{(t)}, a^{(t)})$ over a sufficient period, while adjusting the learning rate γ, $Q(s^{(t)}, a^{(t)})$ is guaranteed to converge at $Q_{\pi^*}(s^{(t)}, a^{(t)})$.

The proposed algorithm framework is summarized in detail in Algorithm 2 as follows: In step 1, from line 1–2, UEs request for admission and initial association via the RRHs from the BBU pool. In step 2, from line 3–10, the Q-table is initialized and updated for each decision epoch as the environment changes in an online learning process. An incoming state space is compared with a general hash function ($H_k(s)$) and the closest is selected as the hash code for the incoming state. For any stage t, the selected hash code is fed into the Q-learning agent as input states. As learning is in process, the agent explores the environment by selecting random actions. The agent chooses an action $a^{(t)}$ at stage t with a probability as shown in Eq. (22). After some time, the agent selects actions based on its past experience by selecting an action with the maximum Q-value. In step 3, in line 12–15, UEs will be re-associated and allocated with resources based on the set of active RRHs for rate in (8) or delay in (9). Lastly, we observe the reward $r^{(t)}$ and the next state $s^{(t+1)}$ of stage t and update the Q-table using Eq. (25). The process is iterated until the Q-table converges.

D. Customized physical resource allocation

The physical resource allocation could be formulated as an optimization problem and an optimal beamforming solution can be derived by maximizing the aggregate QoS satisfaction of UEs. A unified design of physical resource allocation models is given in (26), which could be customized for different operators. The objective function is expressed as;

$$\max_{w} \sum_{i \in I} \eta_i \xi_i(\cdot), \tag{26}$$

such that;

$$\sum_{i \in I} \|w_{ij}\|^2 \leq P^t_{max} \forall j \in J \tag{26a}$$

$$\sum_{u \neq i} \left|\sum_{j \in J} g_{ij} w_{uj}\right|^2 \leq \Gamma_{th} \forall i \in I \tag{26b}$$

where $\eta_i > 0$ is the weighting factor for UE $i \in I$, $\xi(\cdot)$ denotes the satisfaction of UE i, which is customized for rate as $\xi(r_i)$ or delay as $\xi(\tau_i)$, w is the beam forming vector/optimization variable, P^t_{max} represents the maximum transmission power and Γ_{th} is the interference threshold. Constraint (26a) ensures that, the sum of transmit power of UEs does not exceed the maximum transmission power. Constraint (26b) ensures that the sum of the received interference power of UEs does not exceed the interference threshold. For the rate constraint users, the satisfaction on rate follows the form of (8) as $\xi(r_i)$. For the delay constraint users, the satisfaction on delay is in the form of (9) as $\xi(\tau_i)$. The objective of our function is to maximize user satisfaction, which can be transformed into a convex optimization problem known as second order cone optimization problem. The second order cone optimization problem can be solved efficiently using an existing method in [32].

E. Convergence analysis and optimality

In this paper, the Markovian property of the process is investigated in terms of the history of aggregated states, while in practice the Markovian property of interest is in terms of the history of raw observations. Thus, perhaps the more relevant measure for the aggregated Markovian processes investigated here is how much better an outcome (rewards and observations) can be predicted if the agent is given access to the raw state instead of the aggregated state. Q-learning was known to converge to the optimal policy [33]. In this paper, the MDP $\mathcal{M} = \left[S, A, R(s, a), S'\right]$ is formulated with aggregated features via clustering, which is compressed as a set of *anchors*. Let us denote optimal Q-functions for continuous states and discrete states as $Q^*(\bar{s}, a)$ and $Q^*(\tilde{s}, a)$ respectively. In this section, we theoretically prove that the proposed clustering-based Q-learning

Algorithm 2: The revised Q-learning algorithm framework

Request for UEs,
Build initial association among UEs and RRHs;
For each decision epoch t, (UE demand changed by traffic model) **do**
  Calculate the hash code of (s) using the general hash function $H_k(s)$
  Find the closest anchor $(\widetilde{s^*}_m)$ to (s) w.r.t hamming distance
  Feed the RL agent with the hash code of $(\widetilde{s^*}_m)$;
  Select the activation action a randomly using Eqn. (22);
  Otherwise, select the activation action by $a' = argmaxQ(s, a)$ using Eqn. (24);
  Execute activation action a;
  Obtain the set of RRHs J;
  /* Resource Allocation */
  Obtain the optimal beamforming solution based on given a by solving Eqn. (26);
  Calculate the user satisfaction for rate or delay constrained users;
  Execute step 13 using Eqn. (8) or Eqn. (9);
  Reconfigure the network;
  /* Update */
  Observe the reward $R(s, a)$ using new state s';
  Store the state transition (s, a, r, s');
  Update the Q-table using Eqn. (25);
  Iterate the process until the Q-table converges;
end for

algorithm will converge to the optimal solution of Q-learning for the continuous *raw observations*.

Based on the conclusion in reference [34], assuming the compactness and Lipschitz continuity holds, the Q-value function of the continuous dynamic programming and the discrete dynamic programming satisfying

$$\left|Q^{(t)}(\bar{s}, a) - Q^{(t)}\left(\tilde{s}, a\right)\right| < \alpha(\|\bar{s} - \tilde{s}\|), \forall \bar{s}, \tilde{s} \in S, \tag{27}$$

where α is a positive constant. In this way, Eq. (27) can be rewritten as

$$\left|Q^{(t)}(\bar{s}, a) - Q^{(t)}\left(\tilde{s}, a\right)\right| < \beta d_s, \forall \bar{s}, \tilde{s} \in S, \tag{28}$$

where $d_s = \min \|\bar{s} - \tilde{s}\|$ is referred to as the hamming distance between *raw observations* and *clustering centroids*. Then we consider the scenario in long-term stochastic control problem when t approaches infinity.

Theorem 2. *If continuous stochastic optimal problem satisfies the compactness and Lipschitz continuity, for each $s \in S$, $Q^{(t)}(s, a)$ converges to the optimal solution $Q^*(\bar{s}, a)$ as the radius of each cluster small enough and t approaches infinity ($t \to \infty$) i.e.*

$$\lim_{d_s \to 0} P\left\{|Q^{(t)}(s, a) - Q^*(\bar{s}, a)| = 0\right\} = 1. \tag{29}$$

where s is raw observation waiting for clustering. If the original Q-learning converges and $t \to \infty$, $Q^{(t)}(s, a)$ in Eq. (25) also converges to the optimal Q-function $Q^*(\bar{s}, a)$ w.p.1 for any $s \in S$.

Proof. Under the assumption of compactness and Lipschitz continuity holds, $\lim_{d_s \to 0} |Q^*(\bar{s}, a) - Q^*\left(\tilde{s}, a\right)| = 0 \to$ holds. This has been proved in case of uniform state clustering [34] and in case of variable resolution discretization [34]. Therefore, there exists a $\delta > 0$ such that if $d_s < \delta$,

$$|Q^*(\bar{s}, a) - Q^*(\tilde{s}, a)| < \varepsilon_1, \forall \bar{s}, \tilde{s} \in S, a \in A\left(\tilde{s}\right), \forall \varepsilon_1 (0 < \varepsilon_1 < \varepsilon), \tag{30}$$

where $Q^*(\bar{s}, a)$ and $Q^*\left(\tilde{s}, a\right)$ are the optimal Q-values for continuous states and the discrete states.

With the definition of clustering-based Q-learning solution for discrete dynamic programming in the partitioned state spaces, for any $s \in S_i$, where $i = 1, 2 \ldots \cdot, k$, $Q^{(t)}(s, a)$ also converges to $Q^*(\tilde{s}, a)$ w.p.1 with assumptions and conditions, which has been provided in [34] and [35]. Furthermore, it has been shown that convergence is guaranteed no matter how states are sampled [36]. However, it is not guaranteed to converge with the same optimal solution. The latest discussion on this topic can be found in [37] and [38]. If compactness and Lipschitz continuity hold, $\forall \varepsilon_2 (0 < \varepsilon_2 < \varepsilon - \varepsilon_1)$, there exists an M such that if $k > M$, it holds that

$$P\{|Q^{(t)}(s, a) - Q^*\left(\tilde{s}, a\right)| < \varepsilon_2\} = 1, \forall \bar{s}, \tilde{s} \in S \text{ and } a \in A(\tilde{s}). \tag{31}$$

Therefore, from (30), (31), for an arbitrary $\varepsilon > 0$, there exist an M and $\delta = \delta_1 > 0$ such that *if* $d_s < \delta$ and $k > M$, then

$$\begin{aligned} &P\{|Q^{(t)}(s, a) - Q^*(\bar{s}, a)| < \varepsilon\} \\ &\geq P\{|Q^{(t)}(s, a) - Q^*(\bar{s}, a)| + |Q^*(\bar{s}, a) - Q^*(\tilde{s}, a)| < \varepsilon\} \\ &\geq P\{|Q^{(t)}(s, a) - Q^*(\tilde{s}, a)| < \varepsilon - \varepsilon_1\} \\ &\geq P\left\{|Q^{(t)}(s, a) - Q^*(\tilde{s}, a)| < \varepsilon_2\right\} = 1 \end{aligned}$$

Therefore, $P|Q^{(t)}(s, a) - Q^*(\bar{s}, a)| < \varepsilon = 1$, this implies that Eq. (29) holds.

5. Performance evaluation

5.1. Scenario configuration

In this section, the proposed revised Q-learning scheme is compared with the classical pure Q-learning [6], and deep reinforcement learning [8]. We conducted our simulation using Pytorch for DQN and MATLAB for Q-Learning. We focus on three essential performance criteria for evaluation: convergence rate,

Table 2
Simulation Parameters.

Parameter units	Value
Number of RRH, J	3
RRH coverage	200 m
System Bandwidth, B	20 MHz
Maximum transmit power per RRH, P^{l}_{max}	1W
Noise power spectrum density, σ	−174 dBm/Hz
Carrier Frequency band, F	2.4 GHz
Path loss model	32.4 + 20log(F) + 20log(D)
Shadowing effects, δ	0-8 dB, random
Packet arriving rates, λ	160 packets per second
Angular coefficient in traffic model, υ	0.1
Number of UEs in traffic model, I	0-32
UE demand, r^{min}/ r^{max}	0.5 Mbps/50 ms
Steepness coefficient in satisfaction model, η	1e−5
Power consumption(active state), P^{t}_{j}	Eq. (7)
Power consumption(sleep state), P^{s}_{j}	4.3 W
Mean packet size (in bit), L	4000
Weighted factor in reward function, ω	8
Discount factor, γ	0.5
The number of nearest anchors for matrix Z	2
Number of cluster, $\|M\|$	300
Number of bits (Hash code length), r	15
Decision epoch, t	60 min
Number of training samples, $\|U\|$	3e4

total energy consumption and satisfaction of UEs. We evaluate the performance considering the effect of dynamics in user mobility and changes in traffic load. It is assumed that the BBU initially manages a maximum of 3 RRHs. The system bandwidth of RRHs is set at 20 MHz. The threshold of UE sensitivity is set at −120 dBm for edge UEs. In the network model, the RRH coverage in the network is 200 m. The number of UEs ranges from 0 to 32 in each C-RAN according to the traffic model profile [18]. The rate or delay demand is equal for each of the UEs. The 10% of UEs with the lowest signal strength are considered as edge UEs to work in CoMP mode. The energy consumption largely depends on traffic load and the active duration of RRH, based on the energy model for the active and idle states respectively [8]. The value of $\in$ is 0.01 in the $\in$-greedy policy, which is defined as the probability by which an agent takes a random action promoting exploration instead of an action determined by the maximum of the Q-value of the next state. We clarify the typical simulation parameters for system and algorithms as summarized in Table 2.

5.2. Convergence analysis

This subsection compares the convergence rates of the deep Q-network (DQN), pure Q-learning (QL), and the proposed revised Q-learning (AGH+QL) method in this paper. The analysis is performed under static scenario without mobility, traffic changes per hour based on traffic profile model. The reward $R(s, a)$ is deterministic and chosen as the average satisfaction of all users. The values of the input parameters used for numerical tests with the algorithm are lambda (λ), rate r, and rate over lambda (r/λ). From Fig. 3, it can be observed that all algorithms could not well learn the behavior of the system at the very onset and experienced fluctuations in energy consumption and satisfaction. Fig. 3 shows that AGH+Q-Learning consistently performs best, and in most cases significantly outperforms Deep Q-network in all state definitions. It can be observed that DQN converges much slower due the fact that is has to explore a larger environment. AGH+QL converges faster, around day 200, because the state space has been reduced by means of the anchor graph hashing. For energy consumption, it can be noted that AGH+QL converges to near optimal values compared to the DQN. This is due to the fact that our proposed algorithm is able to map the environment to the Q-table refined via anchor graph hashing leading to its higher accuracy. For QL, there exist fluctuations in the curves due to its inability to learn from large states space, with states defined as a load vector and resorts to random policy selection. In general, the plots of Fig. 3(1(a)–1(c)) show improvements in convergence rate when states are defined as rate over lambda (r/λ). When using rate only as compared to lambda only, there is practically no difference in the convergence rate for pure QL and AGH+QL. However, for Deep Q-network when the input is changed from rate to lambda, we see a noticeable difference between the convergence rate, especially in the plots for satisfaction and energy. A visible improvement in AGH+Q-Learning is also evident in satisfaction over the Deep Q- network. Here, there is an obvious benefit in defining rate over lambda (r/λ) as states, as many states in domain will be explored.

5.3. Effect of anchor graph hashing

Fig. 4 shows the energy consumption and satisfaction curves using hash lookup within a hamming radius of 2. When longer codes are used, the radius of clusters will be larger due to increased sparsity of the relational matrix. The increase in bits reduces the accuracy of Q-learning significantly. It can be observed that, energy consumption increases as the length of the hash code increases. We can conclude that inaccurate prediction can lead to poor energy consumption. However, the effect of the number of clusters cannot be ignored. The graphs reveal that the number of clusters has an impact on energy consumption and user satisfaction.

From Fig. 4, it can be observed that, smaller number of clusters consumes a substantially higher amount of energy as the length of the hash code increases. Using the preference of the users, the clustering module was trained to obtain the required centroids. With respect to the length of hash codes, the number of clusters was observed to determine the ideal number of clusters which can achieve minimum energy consumption and highest satisfaction ratios. 300 clustered network tends to have relatively lower energy consumption even for longer codes. However, the performance of 30 and 150 clusters improves rapidly as the hash code length reaches 23 and 26 respectively. We argue that the improved performance is due to the learning algorithm reaching its optimal solution. Therefore, to ensure long term efficient energy consumption, the hash codes should be long enough such that the Q-learning agent can learn to gain better knowledge over the state to find proper resource allocation strategy.

We compare the convergence of AGH+QL on average energy consumption of all anchors with different values of r. Fig. 5 shows all values, r, converge in terms of energy consumption which confirms that the algorithm converges. Also it shows the AGH+QL converges faster. When $r = 5$ and 10, it converges before the 100th simulation time, while values 15 and 20 converges after the 200th simulation time, with values 25 and 30 converging close to the 300th simulation time. This implies that a smaller value of r in AGH+QL reduces the sparsity in the adjacent matrix which preserves the similarity relationship in Hamming space. Furthermore, smaller values of r e.g. 5, 10 and 15 have less fluctuation when they converge. This shows that setting the value of r to as small as possible improves the accuracy of the learning process. The impact on satisfaction as shown in Fig. 5(b) shows a similar pattern in terms of rate of convergence and fluctuations in the curve but all values of r are able to achieve a desirable satisfaction level (above 0.5). The proposed AGH+QL can learn the exact behavior of the network. Therefore, choosing the appropriate length of hash codes can minimize the energy cost and ensure user satisfaction.

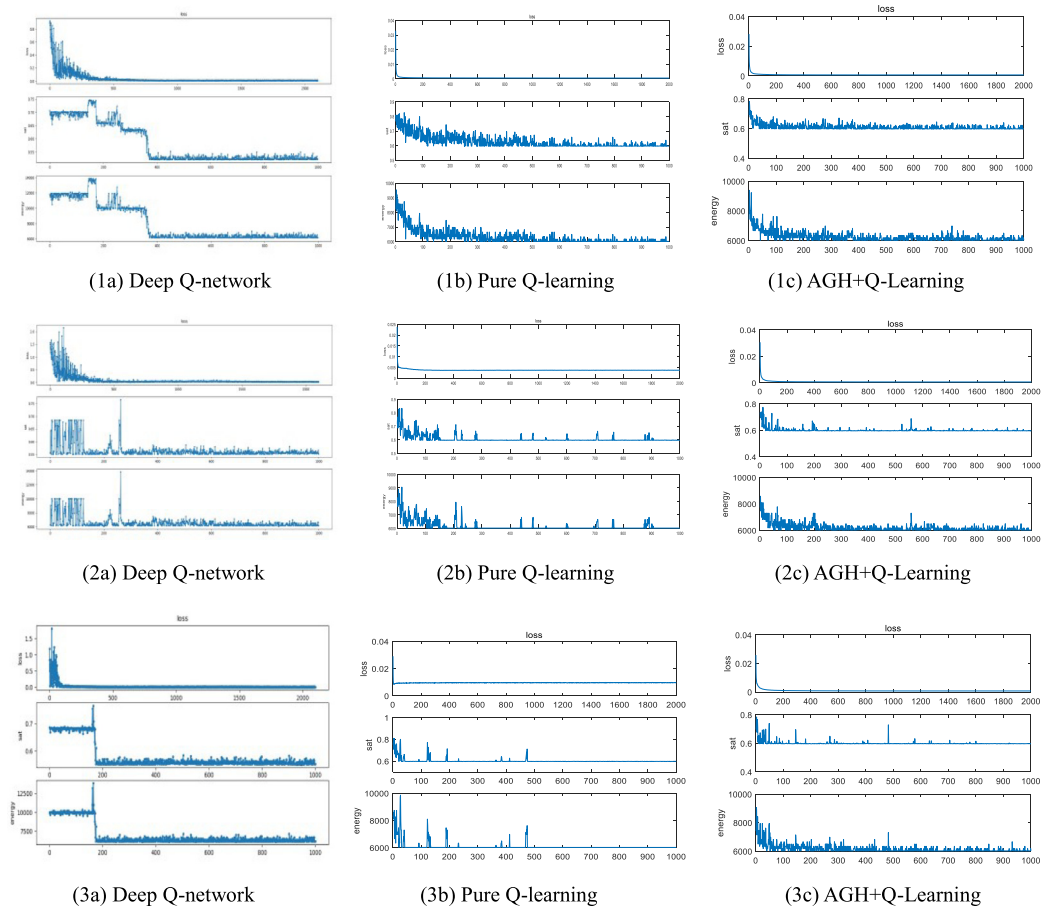


Fig. 3. Energy consumption and satisfaction convergence rate with Lambda (a), Rate (b) and Rate over Lambda (c) as inputs.

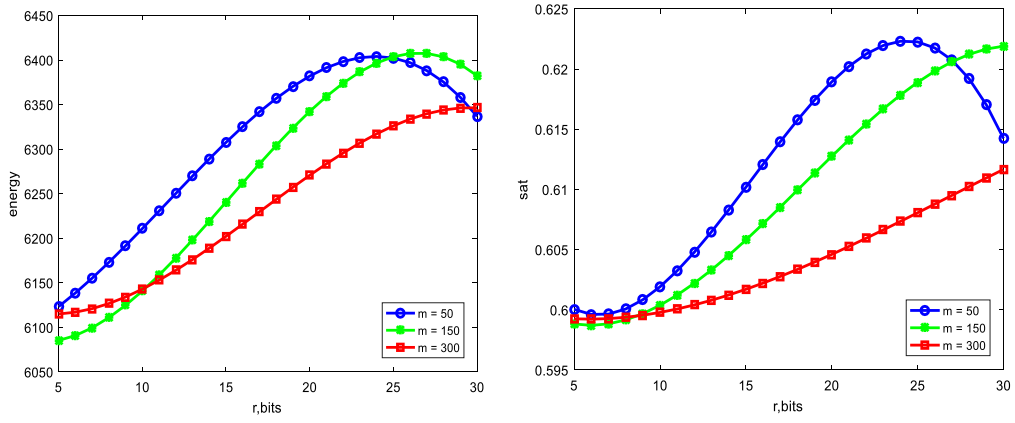
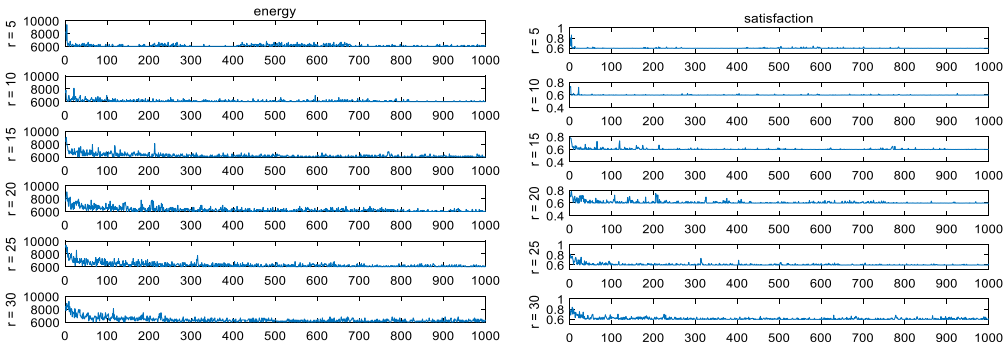


Fig. 4. Performance evaluation using hash lookup with the varying number of hash bits r.

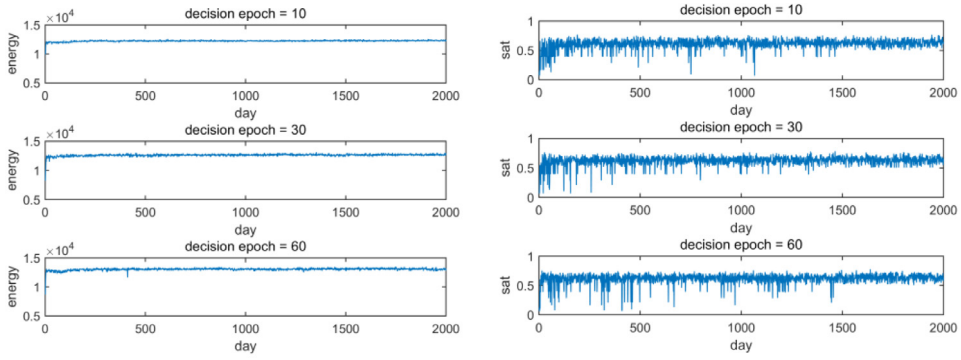
5.4. Effect of the decision epochs

Figs. 6 and 7 show analysis of energy consumption and satisfaction generated by dynamic state given decision epochs 10, 30 and 60 under mobility epoch of 10 min. The users in the system may vary in accordance with traffic model. It should be noted that, the traffic changes every hour based on daily traffic model. Fig. 6 shows the learning performance for the first 2000 days when using a fixed learning step i.e. decision epoch to update the states over a period of time (days). Firstly, we compared with DQN and pure Q-learning(QL). We observed that both algorithms are not able to converge as a result of the varying user mobility. These algorithms represent the traffic load on each RRH as a vector and cannot capture the user mobility information. Furthermore, Figs. 6 and 7 show how energy consumption and satisfaction change with time among decision epoch 10, 30 and 60 under AGH+QL and QL algorithms respectively. During this time, the learning agent makes different decisions as to the number of active RRHs based on the traffic model. We notice that the decision epoch chosen does not have a strong impact on satisfaction. From Fig. 6, the proposed AGH+QL algorithm is able to achieve convergence at all the decision epochs close to



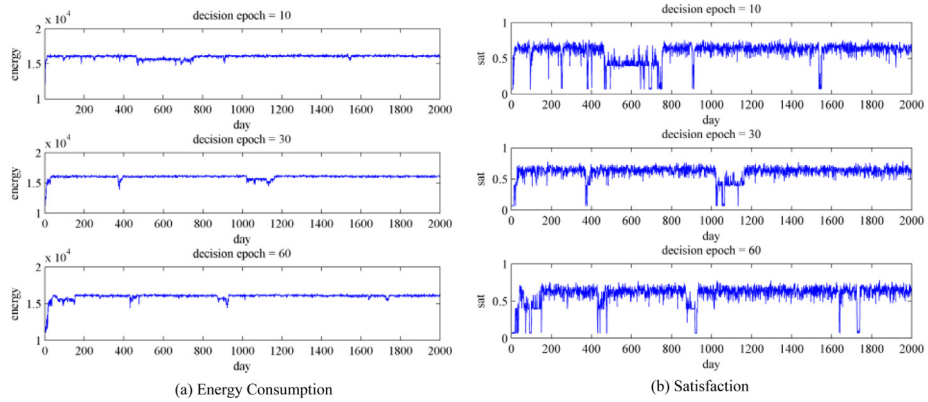
(a). Energy Consumption (b). Satisfaction

Fig. 5. Energy consumption and satisfaction with varying number of hash bits (r).



(a) Energy Consumption (b) Satisfaction

Fig. 6. Impact of decision epoch on convergence of AGH+QL.

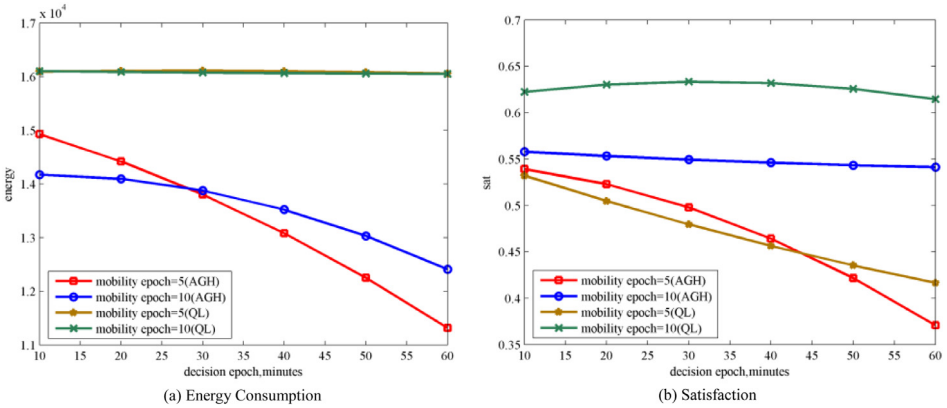


(a) Energy Consumption (b) Satisfaction

Fig. 7. Impact of decision epoch on convergence of QL.

day 1500 with a minimum satisfaction level of 0.5. This shows that, regardless of the decision epoch, the agent is able to capture the user mobility effectively in order to ensure all users are satisfied at any point in time. Given an appropriate decision epoch, AGH+QL can ensure all users are satisfied at acceptable energy consumption. From Fig. 7, it is observed that QL spends more time to converge than our proposed algorithm with a minimum satisfaction level in the last 500 days as 0.4285. QL has a higher energy consumption level than the AGH+QL algorithm under the same conditions. Since QL has more states, it has less training chances for each state and this results in the fluctuations and spikes in the convergence results. It can be concluded that AGH+QL achieves a faster and better convergence than QL under the same conditions.

To demonstrate the robustness of the AGH+QL algorithm, we show the average energy consumption and satisfaction in different mobility epochs for a highly dynamic scenario and a moderate dynamic scenario for both algorithms in the last 500 days. In Fig. 8, we extend the analysis in Figs. 6 and 7 to determine the characteristics of user mobility and analyze the effect it has on decision epoch. Here, we define the decision epoch to be in a range of 10 min to 60 min. We define mobility epoch of 5 min for a highly dynamic scenario and 10 min for a moderate dynamic scenario. It is evident from Fig. 8 that, as the decision epoch increases, the performance of AGH+QL with respect to energy consumption improves whiles the energy consumption levels achieved by QL is very high. At decision epoch of 60 min, the energy consumption level of 10 min mobility epoch for AGH+QL algorithm is approximately 1.25×10^4 and the satisfaction is



(a) Energy Consumption (b) Satisfaction

Fig. 8. Energy consumption and satisfaction vs mobility epoch.

about 0.54. Under the same conditions, the energy consumption and satisfaction levels for QL algorithm are approximately 1.60×10^4 and 0.62 respectively. When the mobility epoch is 5 min, the energy consumption and satisfaction levels for AGH+QL algorithm at the same decision epoch are approximately 1.13×10^4 and 0.37 respectively. For QL, the energy consumption is 1.60×10^4 and its corresponding satisfaction is approximately 0.43. The satisfaction level of mobility epoch 5 falls below 0.5 at about decision epoch 40 and falls further with increasing decision epoch. This means that both AGH+QL and QL maintain the satisfaction of users under moderate dynamic scenarios but fail under highly dynamic scenarios. Under moderate dynamic scenario, AGH+QL achieves a better performance than QL because QL has more states and need more time to update the values in the Q-table. AGH+QL has fewer state space due to clustering and AGH used. Although QL achieves a better satisfaction under moderate dynamic scenario, it consumes more energy compared to AGH+QL. We can conclude that energy saving comes with a cost of some users not being satisfied. The proposed algorithm (AGH+QL) tries to find a better balance between energy consumption and user satisfaction for moderate mobility scenarios. However, both algorithms fail in highly dynamic mobility scenarios because they cannot capture the highly dynamic states of the users.

6. Conclusion

In this paper, we proposed a novel revised Q-learning based framework of autonomous cell activation and customized physical resource allocation schemes for energy consumption and QoS optimization in C-RANs. In the cell activation scheme, we set up a novel anchor graph hashing based Q-learning model to satisfy the QoS requirements of users and to achieve low energy consumption with the minimum number of active RRHs under varying traffic demand and user mobility. The proposed scheme, AGH based Q-learning, was compared with the pure Q-learning, and the recent DQN based schemes. Simulation results showed that, the proposed AGH based Q-learning schemes can converge at a moderate rate, also can capture dynamics for realistic scenarios and achieve a better balance between energy consumption and user satisfaction.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Research Foundation of China, Grant no. 61771098, by the Fundamental Research Funds for the Central Universities, China under grant no. ZYGX2018J049, by the fund from the Department of Science and Technology of Sichuan Province, China, Grant No. 2017GFW0128, 8ZDYF2268, 2018JY0578 and 2017JY0007, and the ZTE Innovation Research Fund, China for Universities Program 2016.

References

[1] N. Bhushan, D. Malladi, J. Li, S. Geirhofer, Network densification: the dominant theme for wireless evolution into 5G, IEEE Commun. Mag. 52 (2) (2014) 82–89.
[2] S. Luo, R. Zhang, T.J. Lim, Downlink and uplink energy minimization through user association and beamforming in C-RAN, IEEE Trans. Wireless Commun. 14 (1) (2015) 494–508.
[3] M. Peng, Y. Li, J. Jiang, C. Wang, Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies, IEEE Wirel. Commun. 21 (6) (2014) 126–135.
[4] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.
[5] B. Zhuang, D. Guo, M.L. Honig, Energy-efficient cell activation, user association, and spectrum allocation in heterogeneous networks, IEEE J. Sel. Areas Commun. 34 (4) (2016) 823–831.
[6] Adnan Aijaz, Hap-SliceR: A radio resource slicing framework for 5G networks with haptic communications, IEEE Syst. J. (99) (2017) 1–12.
[7] J. Ye, Y.A. Zhang, DRAG: deep reinforcement learning based base station activation in heterogeneous networks, 2018, eprint arXiv:1809.02159.
[8] X. Zhiyuan, Y. Wang, J. Tang, A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs, in: IEEE International Conference on Communications (ICC), 2017, pp. 1-6.
[9] S. Mannor, I. Menache, A. Hoze, U. Klein, Dynamic abstraction in reinforcement learning via clustering, in: Proceedings of the 21st International Conference on Machine Learning, ACM, 2004, pp. 71–78.
[10] S. Buzzi, C.L. I, T.E. Klein, H.V. Poor, C. Yang, A. Zappone, A survey of energy-efficient techniques for 5G networks and challenges ahead, IEEE J. Sel. Areas Commun. 34 (4) (2016) 697–709.
[11] Y. Shi, J. Zhang, K.B. Letaief, Group sparse beamforming for green cloud-RAN, IEEE Trans. Wireless Commun. 13 (5) (2014) 2809–2823.
[12] Y. Shi, J. Zhang, W. Chen, K.B. Letaief, Enhanced group sparse beamforming for green cloud-RAN: a random matrix approach, IEEE Trans. Wireless Commun. 17 (4) (2018) 2511–2524.
[13] B. Dai, W. Yu, Energy efficiency of downlink transmission strategies for cloud radio access networks, IEEE J. Sel. Areas Commun. 34 (4) (2016) 1037–1050.
[14] M. Peng, K. Zhang, J. Jiang, J. Wang, W. Wang, Energy-efficient resource assignment and power allocation in heterogeneous cloud radio access networks, IEEE Trans. Veh. Technol. 64 (11) (2015) 5275–5287.
[15] Yicheng Lin, Wei Bao, Wei Yu, Ben Liang, Optimizing user association and spectrum allocation in hetnets: a utility perspective, IEEE J. Sel. Areas Commun. 33 (6) (2014) 1025–1039.
[16] Wei-Sheng Lai, Tsung-Hui Chang, Ta-Sung Lee, Joint power and admission control for spectral and energy efficiency maximization in heterogeneous OFDMA networks, IEEE Trans. Wireless Commun. 15 (2016) 3531–3547.

[17] G.P. Koudouridis, P. H. Gao, A centralised approach to power on-off optimization for heterogeneous networks, in: IEEE Vehicular Technology Conference (VTC Fall), 2012, pp. 3-6.
[18] G. Sun, P.C. Addo, G. Wang, G. Liu, Energy efficient cell management by flow scheduling in ultra-dense networks, KSII Trans. Internet Inf. Syst. 10 (9) (2016) 4108–4122.
[19] X. Zhang, J. Zhang, Y. Huang, W. Wang, On the study of fundamental trade-offs between qoe and energy efficiency in wireless networks, Trans. Emerg. Telecommun. Technol. 24 (3) (2013) 259–265.
[20] A. Mesodiakaki, F. Adelantado, L. Alonso, C. Verikoukis, Energy-efficient context-aware user association for outdoor small cell heterogeneous networks, in: IEEE Int. Conf. on Commun. (ICC), 2014, pp. 1614–1619.
[21] Y. Xu, R. Hu, L. Wei, G. Wu, QoE-aware mobile association and resource allocation over wireless heterogeneous networks, in: IEEE Global Commun. Conf. (GLOBECOM), 2014, pp. 4695–4701.
[22] M. Chen, U. Challita, W. Saad, C. Yin, M. Debbah, Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks, 2017, arXiv:1710.02913.
[23] M. Chen, W. Saad, C. Yin, M. Debbah, Echo state networks for proactive caching in cloud-based radio access networks with mobile users, IEEE Trans. Wireless Commun. 16 (6) (2017) 3520–3535.
[24] M. Chen, W. Saad, C. Yin, Virtual reality over wireless networks: Quality-of-service model and learning-based resource management, IEEE Trans. Commun. 66 (11) (2018) 1562–5635.
[25] A. Moubayed, A. Shami, H. Lutfiyya, Wireless resource virtualization with device-to-device communication underlaying LTE network, IEEE Trans. Broadcast. 61 (4) (2015) 734–740.
[26] G. Auer, O. Blume, V. Giannini, I. Godor, M.A. Imran, Y. Jading, E. Katranaras, M. Olsson, D. Sabella, P. Skillermark, W. Wajda, Energy Efficiency Analysis of the Reference Systems, Areas of Improvements and Target Breakdown, Technical Report D23, ALUD, DOCOMO, EAB, ETH, IMEC, TI, UNIS, 2010, Available online.
[27] M. Deruyck, W. Joseph, L. Martens, Power consumption model for macro-cell and microcell base stations, Trans. Emerg. Telecommun. Technol. 25 (3) (2014) 320–333.
[28] W. Liu, He. J, S.F. Chang, Large graph construction for scalable semi-supervised learning, in: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 679-686.
[29] K.I. Williams, M. Seeger, Using the nystrom method to speed up kernel machines, in: Advances in Neural Information Processing Systems (NIPS), 2000, p. 13.
[30] D.A. Duwaer, On Deep Reinforcement Learning for Data-Driven Traffic Control, LD Software, Eindhoven, 2016.
[31] R. Li, Z. Zhao, X. Chen, J. Palicot, H. Zhang, TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks, IEEE Trans. Wireless Commun. 13 (4) (2014) 2000–2010.
[32] M.S. Lobo, L. Vanderberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebr. Appl. J. 284 (1) (1998) 193–228.
[33] C.J.C.H. Watkins, Learning from Delayed Rewards, (Ph.D. dissertation), King's college, UK, 1989.
[34] D. Bertsekas, Convergence of discretization procedures in dynamic programming, IEEE Trans. Automat. Control 20 (3) (1975) 415–419.
[35] R. Munos, A. Moore, Variable resolution discretization in optimal control, Mach. Learn. 49 (2–3) (2002) 291–323.
[36] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, MA, 1996.
[37] D.P. Bertsekas, Feature-based aggregation and deep reinforcement learning: A survey and some new implementations, IEEE/CAA J. Autom. Sin. 6 (1) (2019) 1–31.
[38] M. Hutter, Extreme state aggregation beyond MDPs, in: International Conference on Algorithmic Learning Theory (185-199), Springer, Cham, 2014.

Guolin Sun received his B.S., M.S. and Ph.D. degrees all in Communication and Information Systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2000, 2003 and 2005 respectively. After Ph.D. graduation in 2005, Dr. Guolin has got eight years industrial work experience on wireless research and development for LTE, Wi-Fi, Internet of Things (ZIGBEE and RFID, etc.), Cognitive radio, Localization and navigation. Before he joined the School of Computer Science and Engineering, University of Electronic Science and Technology of China as an Associate Professor in Aug. 2012, he worked in Huawei Technologies Sweden. Dr. Guolin Sun has filed over 40 patents, and published over 40 scientific conference and journal papers, acts as TPC member of conferences. Currently, he serves as a vice-chair of the 5G oriented cognitive radio SIG of the IEEE (Technical Committee on Cognitive Networks (TCCN)) of the IEEE Communication Society. His general research interest is software defined networks, network function virtualization, advanced radio resource management.

Tong Zhan is an undergraduate student and currently studying B.Eng Digital Media Technology in the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He is also a member of the Undergraduate Research Program (URP) – UESTC in 2018, being a research assistant at the Mobile Cloud-Net Research lab. His research interests include deep reinforcement learning, network resource management, and computer vision.

Gordon Owusu Boateng received his Bachelor degree in Telecommunications Engineering from the Kwame Nkrumah University of Science and Technology, Kumasi-Ghana, West Africa, in 2014. He is currently studying MSc. Computer Science and Technology in University of Electronic Science and Technology of China (UESTC). From 2014 to 2016, he worked under sub-contracts for Ericsson (Ghana) and TIGO (Ghana). He is also a member of the Mobile Cloud-Net Research Team — UESTC. His interests include Mobile/Cloud Computing, 5G Wireless Networks, Data Mining, D2D communications and SDN.

Daniel Ayepah-Mensah received his Bachelor in Computer Engineering from Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, in 2014. He is currently Studying MSc. Computer Science and Technology in University of Electronic Science and Technology of China (UESTC). From 2014 to 2017, he worked as a Software Developer. He is also a member of the Mobile Cloud-Net Research Team — UESTC. His interest includes generally Wireless Networks, Big Data and Cloud Computing.

Guisong Liu received his B.S. degree in Mechanics from the Xi'an Jiao Tong University, Xi'an, China, in 1995, and his M.S. degree in Automatics, Ph.D degree in Computer Science from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and 2007, respectively. Dr. Liu was a visiting scholar at Humbolt University at Berlin during September to December in 2015. Now, he is a full professor in the School of Computer Science and Engineering, the University of Electronic Science and Technology of China, Chengdu, China. He is also the dean of the School of Computer Science, Zhongshan Institute, UESTC, Zhongshan, China. His research interests include pattern recognition, neural networks, and machine learning.

Wei Jiang received his Ph.D degree from Beijing University of Posts and Telecommunications (BUPT) in 2008. In March 2008, he worked for 4 years in Central Research Institute of Huawei Technologies, in the field of wireless communications and 3GPP standardization. In September 2012, he joined the Institute of Digital Signal Processing, University of Duisburg-Essen, Germany, where he was a Postdoctoral researcher and worked for EU FP7 ABSOLUTE project and H2020 5G-PPP COHERENT project. Since October 2015, he joined the Intelligent Networking Group, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany, as a senior researcher and works for H2020 5G-PPP SELFNET project. Meanwhile, he also works for the Department of Electrical and Information Technology (EIT), Technische University (TU) Kaiserslautern, Germany, as a senior lecturer. He served as a vice Chair of IEEE TCCN special interest group (SIG) "Cognitive Radio in 5G". He is the author of more than 30 papers in top international journals and conference proceedings, and has 27 patent applications in wireless communications, most of which have already been authorized in China, Europe, United States or Japan. He wrote a chapter "From OFDM to FBMC: Principles and Comparisons" for the book "Signal Processing for 5G: Algorithms and Implementations" (Wiley, 2016). His present research interests are in digital signal processing, multi-antenna technology, cooperative communications, 5G, and machine learning.