entropy

MDPI

Article

A Turbo Q-Learning (TQL) for Energy Efficiency Optimization in Heterogeneous Networks

Xiumin Wang [1], Lei Li [1], Jun Li [2,*] and Zhengquan Li [3]

[1] College of Information Engineering, China Jiliang University, Hangzhou 310018, China; 05a0303091@cjlu.edu.cn (X.W.); P1803085214@cjlu.edu.cn (L.L.)
[2] Binjiang College, Nanjing University of Information Science & Technology, Wuxi 214105, China
[3] College of Internet of Things, Jiangnan University, Wuxi 214000, China; lzq722@jiangnan.edu.cn
* Correspondence: 07a0303105@cjlu.edu.cn

Received: 15 July 2020; Accepted: 24 August 2020; Published: 30 August 2020

Abstract: In order to maximize energy efficiency in heterogeneous networks (HetNets), a turbo Q-Learning (TQL) combined with multistage decision process and tabular Q-Learning is proposed to optimize the resource configuration. For the large dimensions of action space, the problem of energy efficiency optimization is designed as a multistage decision process in this paper, according to the resource allocation of optimization objectives, the initial problem is divided into several subproblems which are solved by tabular Q-Learning, and the traditional exponential increasing size of action space is decomposed into linear increase. By iterating the solutions of subproblems, the initial problem is solved. The simple stability analysis of the algorithm is given in this paper. As to the large dimension of state space, we use a deep neural network (DNN) to classify states where the optimization policy of novel Q-Learning is set to label samples. Thus far, the dimensions of action and state space have been solved. The simulation results show that our approach is convergent, improves the convergence speed by 60% while maintaining almost the same energy efficiency and having the characteristics of system adjustment.

Keywords: energy efficiency; HetNets; eICIC; Q-Learning; reinforcement learning; multistage decision process

1. Introduction

With the dramatic growing number of wireless devices, more stringent requirements are put forward for performance and energy efficiency of heterogeneous networks (HetNets) [1]. With the increasing complexity of HetNets, the optimization of energy efficiency has more and more challenges and is one of the hot spots of communication network research, especially for the HetNets with 5G BSs. Therefore, in this paper, the efficiency of resource allocation algorithm is studied, in which the reinforcement learning (RL) is utilized and the parameters such as ABS (almost blank sub-frame), CRE (Cell range expansion), and SI-SBSs (sleeping indicaton of small BSs) are jointly considered simultaneously to optimize the energy efficiency of the whole network.

In general, the optimization problems with multi-variables are non-convex NP problems, it is hard to be solved directly. Some can be processed by dividing the original problem into sub-problems which can be iteratively solved with an acceptable complexity. Baidas et al. [2] jointly considered subcarriers assignment and global energy-efficient (GEE) power allocation, and the original problem was divided into two subproblems as subcarrier allocation by many to many matching and GEE maximizing power allocation. By designing a two-stage solution program, the original problem was effectively solved with the ensured stability. Chen et al. [3] jointly investigated the task allocation and CPU-cycle frequency, in order to achieve the minimum energy consumption which scaled down to the

sum of two deterministic optimization subproblems by Lyapunov optimization theory. The optimal solutions of the two sub-problems separately which were local computation allocation (LCA) and offloaded computation allocation (OCA) were found to obtain the optimal solution of the upper bound of the original problem. Although decomposition and iteration are efficient to solve the non-convex NP problems in many cases, the complexity of modeling and computing is still high in most cases.

As the AI technologies are developing with a very high speed in the recent years, some learning methods are introduced to solve some complicated optimization problems. As shown in [4–9], model-free RL methods can be an efficient way to solve the energy efficiency optimization problem of HetNets, since the precise model process was not necessary. In [4,5], the Actor–Critic (AC) algorithm was applied to optimize energy efficiency of HetNets while the authors did not conduct in-depth research on the selections of basis functions which are challenging for the application of RL. Roohollah et al. [6] introduced a Q-Learning (QL) based distributed power allocation algorithm (Q-DPA) as a self-organizing mechanism to solve the power optimization problem in the networks. In [7], a method based on QL was proposed to solve the energy efficiency and delay problems of smart grid data transmission in HetNets, in which, however, the dimension of action and state space was too large. Ayala-Romero et al. [8,9] combined dynamic programming, neural networks, and data-driven methods to solve problem of energy saving and interference coordination in HetNets.

In this paper, inspired from the previous works [2,4–10], referring to RL and the idea of converting non-convex NP hard problem into several sub-problems, a turbo QL (TQL) scheme is proposed to optimize energy efficiency in which the traditional QL algorithm is decomposed into several sub-Q-Learning algorithms and has a loop iteration structure, each sub-Q-learning solving each sub-problem. In our scheme, the parameters ABS, CRE, and SI-SBSs are jointly taken into account as action vectors, the user positions are taken as the states in order to fully consider the randomness of users in BSs, and the reward function is designed as a negative reciprocal of the system energy efficiency. The problem of dimensional explosion with increased action space is solved by our proposed TQL structure, and it is acceptable for the complexity of the algorithm. For the states, a fully connected deep neural network is designed to identify state type. The contributions of this paper are summarized as follows.

(1) The reward function is designed as a negative reciprocal of the system energy efficiency to avoid the slow speed of convergence and the possibility of fulling into local optimum. If the magnitude of the reward in RL is too large, it is easy to fall into the local optimization, and too small value can cause the problem of system oscillation or slow speed of convergence. In this paper, directly using energy efficiency as the reward function causes the reward value too large and it is easy to fall into the local optimum. As shown in our experimental results, our designed reward function works well.

(2) The TQL is proposed by combining traditional Q-Learning and multistage decision process which has a loop iteration structure, each sub-Q-Learning solving each sub-problem which is from an original optimization problem. It effectively deals with the dimensional explosion problem caused by the action space increasing in RL and greatly reduces the complexity of optimization problems.

(3) The relevant parameters of each sub-problem can be adjusted independently. Thus, the complexity is low in our proposed TQL algorithm. Simulation results show that the TQL algorithm can solve the original problem with efficiency and flexibility.

The rest of the paper is organized as follows. Related works are summarized in Section 2. Section 3 introduces the system model. In Section 4, the energy efficiency model is formulated. Our proposed algorithm is presented in Section 5. Section 6 shows the simulation framework and numerical results, and conclusions are drawn in Section 7.

2. Related Works

According to the method of solving the problem of resource optimization in HetNets, related works are mainly classified into three aspects as traditional optimization methods, machine learning based approaches, and neural-network-based ones.

2.1. Traditional Methods for Optimizing Hetnets

For a cluster sleeping method, Chang et al. [11] utilized a genetic algorithm to achieve dynamic matching of energy consumption and Li et al. [12] proposed a Gauss–Seidel method to optimize resources in HetNets. In [13], a low complexity algorithm based on the many-to-many matching game between the virtual SMSs, and the users were proposed to solve the problems of exponential growth of mobile data traffic and energy saving. Anany et al. [14] also utilized matching game and proposed an association algorithm which jointly considered the rate and power of each wireless device to get optimal association between the wireless devices and the best BSs according to a well-designed utility function. Wang et al. [15] modeled the location of each layer of BS as an independent Poisson Point Process (PPP) to analyze the coupling relationship between the probability of successful transmission and BS activation. Dong et al. [16] adopted the Poisson clustering process (PCP) method and analyzed the local delay of discontinuous transmission (DTX) mode. For correctly deploying and expanding HetNets to avoid co-channel interference (CCI), Khan et al. [17] proposed a new three-sector three-layer frequency division multiplexing technology (FFR-3SL). Tiwari et al. proposed a Bayesian minimum mean-square-error (MMSE)-based method to estimate user velocity in [18] and presented a handover-count based minimum-variance-unbiased [19].

2.2. Machine Learning for Optimizing Hetnets

Chang et al. [20] utilized a dynamic programming method to optimize spectrum resources between eNBs and low power consumption nodes (LPNs). Deb et al. [21] presented a measurement data-driven machine learning mode LeAP for power control of LTE uplink interference. Chen et al. [22] put forward a method based on hypergraph clustering to solve the serious accumulated interference and improve the system throughput under the requirement of users' fairness. Different from [8,9], Siddavaatam et al. [23] investigated an energy-aware algorithm based on ant colony optimization. Huang et al. [24] proposed an algorithm based on cross entropy (CE) by a sampling approach to address the problem of user association in an iterative mechanism. Castro-Hernandez et al. [25] proposed the application of clustering algorithm and data mining technology to identify the edge users. Castro-Hernandez et al. [26] also presented clustering algorithm and data mining technology to allow the BS to learn and recognized the received signal strength value autonomously which was from the users' reports in the handover (HO) process. Like [26], according to the trigger condition of user handoffs in BS, Yao et al. [27] proposed that the minimum numbers of user handoffs was transformed into the volume of transmitted data in a certain period of time as a reward function, and then the historical information for volume of transmitted data was used to approximate expectation for the volume of transmitted data by Monte Carlo algorithm. Different from the action based on our proposed scheme, Fan et al. [28] proposed to decompose the QL based on state composed of user-BS into two QLs based on the state of user and BS.

2.3. Neural Networks for Optimizing Hetnets

Many researchers focused on solving heterogeneous network problems with neural networks. Different from Refs. [4,5], Li et al. [29] use convolutional neural networks (CNN) and deep neural network (DNN) network structure as actor part and critic part of the AC algorithm to optimize heterogeneous network resources. Chai et al. [30] proposed an access network modeling and an adaptive parameter adjustment algorithm based on a neural network model. The algorithm was applied to the source and destination switching networks, and the input parameters of users in HetNets were dynamically adjusted according to the required QoS. Fan et al. [31] proposed a fuzzy neural network based on RL to optimize antenna tilt and power to achieve automatic collaborative optimization of power and antenna tilt. Self organizing network entities used cooperative Q-learning and reinforcement back-propagation methods to obtain and adjust their optimization experience to achieve cooperative learning. Similar to [29], some schemes combining neural network with

reinforcement learning ware proposed in [32,33]. Since traditional iterative optimization methods, whether optimal or heuristic, usually needed a lot of iterations to achieve satisfactory performance, and led to considerable computational delay, from the perspective of in-depth learning, Lei et al. [32] proposed a feasible cache optimization method which was to train the optimization algorithms through a DNN in advance, instead of directly applying them in real-time caching or scheduling. The computational burden was transferred to a DDN training phase to reduce the complexity of delay sensitive operation phase significantly. Considering the complexity of base station power optimization in multi-layer heterogeneous networks, Sun et al. [33] proposed a dynamic pico-cell base stations (PBS) operation scheme based on CNN, which dynamically changed the on/off state of PBS according to the user's real-time position, thus reducing the total power of BSS.

As shown in the above analysis, with the increasing complexity of heterogeneous network, more and more parameters are needed to be jointly considered to optimize the network system. It is more difficult to directly solve the network optimization problem by using the traditional optimization scheme. Recently, machine learning technologies have become popular and are applied to the optimization of heterogeneous network resources. Model-free learning brings the convenience of solving non convex problems. However, the acquisition of samples, the design of data labels, and the establishment of Markov decision-making process are all great challenges. In this paper, our proposed TQL algorithm which combines Model-free QL with multistage decision process to optimize the allocation of network resources.

3. System Model

We consider a two-layer HetNets scenario as shown in Figure 1, in which a cell contains the macro base stations (MBSs) and SBSs. The SBSs are randomly deployed within the coverage of MBSs. The sets of the SBSs and the MBSs are denoted as S and M, respectively. The users (UEs) randomly enter the cell. According to a set of UEs association with BSs, UEs are divided into SBS UEs (SUEs) and MBS UEs (MUEs) who are associated with SBSs and MBS, respectively.

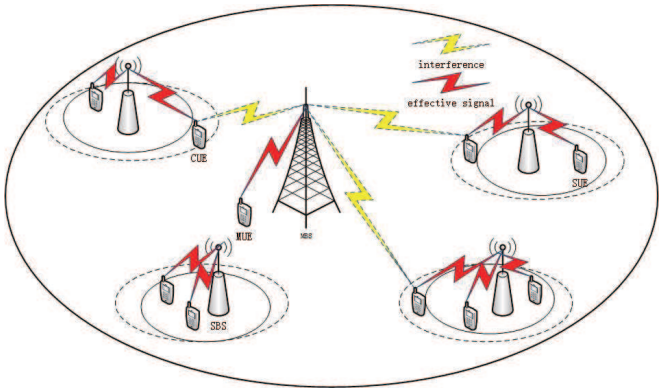

Figure 1. HetNets system model.

In order to balance the load of the entire system network and reduce cross-layer interference by offloading the users of MBSs to SBSs, the enhanced Inter-cell Interference Coordination (eICIC) technology was proposed with two important parameters as ABS and CRE according to [34]. To reduce signal interference, MBSs and SBSs use radio resources in different time periods (subframes) according to eICIC. A frame is divided into some sub frames as ABS and non-ABS subframes, and MBSs normally transmit normal power in nABS (non-ABS) subframes and keep silent or transmit low power in ABS subframes, where the ratio of ABS in a frame is donated as α. The SBSs keep normal transmit power in the whole frames. In the time domain, since the MBSs are allowed to be muted in an ABS subframe period, the interference of the MBSs to the users serviced by SBSs is reduced. Therefore, the SINR

of UEs with poor channel condition is improved since there is no interference from MBSs in these ABS subframes.

In general, the power of MBSs is much higher than that of SBSs, and some UEs should be accessed to the MBSs according to the reference signal receiving power (RSRP). This is because the UEs in LTE networks are associated with the BSs based on RSRP policy where the UEs are connected to the BSs with the highest reference signal. To balance load and improve the system capacity, SBSs are designed to enhance the frequency multiplexing of the network. CRE was proposed to support SBSs to extend their coverage by adding a bias to their RSRP in which users outside the edge of the SBSs can be connected to the SBSs. The UEs located in the extended area of the SBSs receive less interference from MBSs in ABS subframes and get better channel gain to improve their SINR.

Due to the two operating modes of the MBSs in the ABS subframes and non-ABS subframes (nABS), there are also two interference modes for the UEs in the downlink in the HetNets. When the MBSs are in the ABS subframes, the UEs receive only the signal transmitted by the SBSs. However, the MBSs are in the nABS subframes, and the UEs are interfered by the transmitted signal from the SBSs and the MBSs.

The $SINR_{k,n}$ of the UE n connected to the MBS k can be expressed as

$$SINR_{k,n} = \begin{cases} \dfrac{P_M^{k,n} G_{k,n}}{\sum\limits_{j \in M, j \neq k} P_M^{j,n} G_{j,n} + \sum\limits_{i \in S} P_S^{i,n} G_{i,n} + N_0} & k \in M^{nABS} \\ \dfrac{P_m^{k,n} G_{k,n}}{\sum\limits_{j \in M, j \neq k} P_m^{j,n} G_{j,n} + \sum\limits_{i \in S} P_S^{i,n} G_{i,n} + N_0} & k \in M^{ABS} \end{cases} \tag{1}$$

where $P_M^{k,n}$ is the transmission power from the MBS k to the UE n in the nABS subframes, $P_m^{j,n}$ is the transmission power from the MBS j to the UE n in the ABS subframes, $G_{k,n}$ represents the channel gain from the MBS k to UE n, $P_S^{j,n}$ denotes the transmission power from the SBS j to the UE n, N_0 indicates noise variance of the additive white Gaussian, M^{ABS} and M^{nABS} are denoted as MBSs in the ABS and nABS subframes periods, respectively. Note that m and M are short for M^{ABS} and M^{nABS}, respectively.

The $SINR_{k,n}$ of the UE n connected in the SBS k can be written as

$$SINR_{k,n} = \begin{cases} \dfrac{P_S^{k,n} G_{k,n}}{\sum\limits_{j \in M, j = k} P_M^{j,n} G_{j,n} + \sum\limits_{i \in S, i \neq k} P_S^{i,n} G_{i,n} + N_0} & k \in S^{nABS} \\ \dfrac{P_S^{k,n} G_{k,n}}{\sum\limits_{j \in m, j = k} P_m^{j,n} G_{j,n} + \sum\limits_{i \in S, i \neq k} P_S^{i,n} G_{i,n} + N_0} & k \in S^{ABS} \end{cases} \tag{2}$$

where S^{ABS} and S^{nABS} are denoted as the SBSs sets in the ABS subframes and nABS subframes, respectively. Hence, the transmission rate of UE n connected to the BS k can be given as

$$R_{k,n} = \begin{cases} (1-\alpha)\, Blog(1 + SINR_{k,n}) & k \in M^{nABS} \cup S^{nABS} \\ \alpha Blog(1 + SINR_{k,n}) & k \in M^{ABS} \cup S^{ABS} \end{cases} \tag{3}$$

where B is the system bandwidth.

4. Problem Formulation

For comprehensively optimizing the energy efficiency of HetNets, the parameters such as SI-SBSs, ABS, and CRE should be jointly considered. The optimization problem is modeled by setting the

energy efficiency of the system as the optimization objective function. Based on the above analysis, we can establish a joint optimization energy efficiency problem as

$$\begin{aligned} \max_{\left\{ \begin{smallmatrix} x_{k,n},\alpha \\ P_k \end{smallmatrix} \right\}} & \sum_{k\in M\cup S} \sum_{u\in U} x_{k,n} \log\left(\frac{R_{k,n}}{\sum_{k\in M\cup S} P_k}\right) \\ s.t. \quad & 0 \le x_{k,n} \\ & (1) \sim (3) \end{aligned} \tag{4}$$

where the relationship during state $x_{k,n}$, the CRE setting size of the SB k and transmission power $P_{k,n}$ is closely related. $\sum_{k\in M\cup S} P_k$ is closely related to the number of active SBSs.

Let $x_{k,n}$ represent the connection status between BS k and UE n, which is expressed as

$$x_{k,n} = \{0,1\} \qquad \forall k \in M \cup S, n \in U \tag{5}$$

$$\sum\nolimits_{k\in M\cup S} x_{k,n} = 1 \qquad \forall n \in U \tag{6}$$

where $x_{k,n} = 1$ indicates that a connection is established between BS k and UE n, otherwise 0. Equation (6) represents that each UE in the cell can only be connected to one BS.

The transmission power from the BS k to the UE n at different subframe times can be expressed as

$$P_{k,n} = \begin{cases} P_M^{k,n} & k \in M^{nABS} \\ P_m^{k,n} & k \in M^{ABS} \\ P_s^{k,n} & k \in S \end{cases} \tag{7}$$

where

$$P_M^{k,n} = \cdot N_{TRX} \cdot (P_0^m + R^k \cdot P_{\max}^m) \tag{8}$$

$$P_m^{k,n} = N_{TRX} \cdot P_0^m \tag{9}$$

where N_{TRX} is the number of BS transceivers, P_0^m indicates MBSs consumption power in sleep state, and $P_{\max}^m$ represents maximum transmission power of the MBSs, $R^k \in [0,1]$ denotes the load factor of the BSs, which depends on ABS, CRE, and the load density of the BSs, and $P_S^{k,n}$ is

$$P_S^{k,n} = e^k \cdot N_{TRX} \cdot (P_0^s + R^k \cdot P_{\max}^s) + (1 - e^k) \cdot N_{TRX} \cdot P_{sleep}^s + \Delta \tag{10}$$

where e^k represents active state which is 1, and otherwise 0, $P_{\max}^s$ denotes the maximum RF output power of the SBSs, and P_0^s indicates the power when there is no RF of SBSs, P_{sleep}^s is the power consumption when the BSs transceiver station are in sleep state, and Δ is

$$\Delta = \begin{cases} \varphi \cdot P_0 & \text{When the base station k switches from} \\ & \text{the sleep state to the active state.} \\ 0 & \text{Others} \end{cases} \tag{11}$$

where φ represents the proportion of the BSs that wake up the transceiver from sleep to activation state.

Note that ABS, CRE, and the number of active SBSs all affect the load factor of the BSs, which makes problem (4) become complicated and be a non-convex problem. In order to fully consider the complexity and unknown characteristics of the real environment, the optimization problem (4) can be changed as

$$\begin{aligned} \max_{\left\{ \begin{smallmatrix} \alpha,\beta \\ |S| \end{smallmatrix} \right\}} f(\alpha,\beta,|S|) = & \max_{\left\{ \begin{smallmatrix} x_{k,n},\alpha \\ P_k \end{smallmatrix} \right\}} \sum_{k\in M\cup S} \sum_{u\in U} x_{k,n} \log\left(\frac{R_{k,n}}{\sum_{k\in M\cup S} P_k}\right) \\ s.t \quad & 0 \le x_{k,n} \\ & (1) \sim (11) \end{aligned} \tag{12}$$

where f function is unknown, β represents the CRE parameter, $|S|$ represents the number of SBSs activations.

5. Solution with a Tql Algorithm

It is difficult to solve (12) directly because the complexity of the target optimization system is an unknown non-convex problem. In [12], the Gauss–Seidel method needs too much prior knowledge, which is not as convenient as reinforcement learning. In this paper, the table reinforcement learning method QL is used to optimize the system energy efficiency and then our TQL algorithm is proposed to optimize it.

5.1. Q-Learning Algorithm

The environment is typically formulated as a finite-state Markov Decision Process (MDP) and we set a finite discrete time series $t \in \{0, 1, ..., \infty\}$. ABS and CRE are denoted by $\alpha \in A$ and $\beta \in B$, respectively. The activation state of the SBSs in the state u_t is $e_t \in E$, where $E = (0, 1)^{|P|}$, and $|P|$ represents the number of SBSs in a cell. According to the Control Space Augmentors (CSA) concept mentioned in [9], the SBSs states e_t can be derived based on the number of SBSs activations $|S|$ in the cell, where A, B, and $|S|$ represent a limited set of all parameter configurations. Let $\mathcal{S}$ be a discrete set of environment states and $\mathcal{A}$ be a discrete set of actions. At each step t, the agent senses the environment state $s_t = s \in \mathcal{S}$ and selects an action $a_t = a \in \mathcal{A}$ to be performed, where s is the position of a certain number of UEs in the cell, $\mathcal{S}$ represents the set of cell UEs positions, a is optimal α, β and $|S|$ parameter configurations to optimize the energy efficiency of the system, and $\mathcal{A}$ represents the set of parameter configurations. As a result, the environment makes a transition to the new state $s_{t+1} = s' \in \mathcal{S}$ according to probability $\mathcal{P}(s_{t+1} = s' | s_t = s, a_t = a)$ and thereby generates a reward $r_t = r(s_t, a) \in \mathcal{R}$ passing to the agent. MDP is denoted as a tople $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where

- $\mathcal{S}$ is the set of finite state space;
- $\mathcal{A}$ is the set of finite action space;
- $\mathcal{P}$ is the set of transition probabilities;
- $\mathcal{R}$ represents the set of reward funtion.

(1) State: The position of users at step t is considered as state $s_t = s$, and the set of states is denoted as $\mathcal{S}$.

(2) Action: The ABS configuration α, the CRE bias β, and the number of SBS activations $|S|$ are considered as action $a_t = (\alpha, \beta, |S|)$ at state s_t where $\alpha \in$ A, $\beta \in$ B and $|S| \in \{0, 1, ..., |P|\}$. The action space size is $X = (|P| + 1) \times |\mathrm{A}| \times |\mathrm{B}|$.

(3) State transition: The location of users in the cell changes is considered irregularly, and the state transition is random.

(4) Reward function: The optimization problem is system energy efficiency which is used as reward function, but, in the actual simulation process, the reward is too large, which causes the system to fall into the local optimum easily. Our proposed solution is that negative reciprocal of energy efficiency is designed as the reward.

The goal of RL is to find out the expectation of the strategy with the greatest cumulative reward, which can be expressed as

$$\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})\right] \tag{13}$$

where discount factor γ indicates the degree of influence of successor states on current state, and $r(s_t, a_t, s_{t+1})$ represents the reward of state s_t selecting a_t and then transiting to state s_{t+1}.

The best decision sequence of MAP is solved by the Bellman equation. The state-action value function $q(s, a)$ can evaluate the current state. The value of each state-action is not only determined by the current state but also by the successor states. Therefore, the state-action value function $q(s, a)$ of

the current s can be obtained by the cumulative reward expectation of the state. Bellman's equation can be given as [35]

$$q_\pi(s,a) = E_\pi\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ...|a_t = a, s_t = s\right], \tag{14}$$

which is also equivalent to

$$q_\pi(s,a) = E_\pi\left[r_{t+1} + \gamma q_\pi(s_{t+1}, a_{t+1})|a_t = a, s_t = s\right] \tag{15}$$

Optimal action-value function $Q^*(s,a) = \max_\pi Q^*(s,a)$ can be written as

$$Q^*(s,a) = \sum_{s'} \mathcal{P}(s'|s,a)(r(s,a,s') + \gamma \max_{a'} Q^*(s',a')) \tag{16}$$

The update process of Q-value using a time difference method is expressed as [35]

$$Q(s,a) \leftarrow Q(s,a) + \lambda\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{17}$$

where λ is the learning rate. According to the Formula (17), the QL algorithm is utilized to solve problem (12) as shown in Algorithm 1:

Algorithm 1: The QL for optimizing original problem

Require: the set of state K, the set of action X, earning rate λ_{QL}, greedy probability $\varepsilon_{\mathrm{QL}}$, discount factor γ_{QL}, and $threshold_{\mathrm{QL}}$.
Ensure: Q table.
1: Initialize $Q(s,a)$, state s and n=0 and setting $threshold_{\mathrm{QL}}$;
2: **while** $n <= threshold_{\mathrm{QL}}$ **do**
3: In state s, select the optimal action a with greedy probability $\varepsilon_{\mathrm{QL}}$;
4: Observe r;
5: randomly transfer from s to s';
6: Update $Q(s,a)$ according to Formula (17);
7: $s \leftarrow s'$;
8: $n = n + 1$;
9: **end while**
10: Output: Q table;

5.2. Tql Algorithm

If QL is directly utilized to solve the original problem (12) as shown in Algorithm 1, we can see the action size $X = (|P| + 1) \times |\mathrm{A}| \times |\mathrm{B}|$ is too large, where $|\bullet|$ represents a cardinality of set. Since the action is represented by a vector of three dimensions, the optimization problem can be decomposed into three subproblems. We propose the TQL algorithm which decomposes the objective optimization problem into three sub-problems as optimizing the ratio of ABS α, CRE bias β, and the number of SBS activations $|S|$ to reduce action space size.

5.2.1. Sub-Problem A: Given the Cre Bias β and the Number of Sbs Activation $|S|$ for Optimizing the Abs Ratio α

The action of the sub-problem A is $a_t = (\alpha, \beta, |S|)$ where β and $|S|$ are given. $\alpha \in$ A and the action space size is $|A|$. State $s = s_t$. The tabular method of Q-learning can be used to solve sub-problem A and the updating rule of sub-Q-value can be written as

$$Q_\alpha(s,a) \leftarrow Q_\alpha(s,a) + \lambda_1\left[r + \gamma_1 \max_{a'} Q_\alpha\left(s',a'\right) - Q_\alpha(s,a)\right] \tag{18}$$

and shown in Algorithm 2.

Algorithm 2: The CRE bias β and the number of SBSs activation $|S|$ are given to optimize the ABS ratio α.

Require: $A_t = (\alpha, \beta, |S|)$.
Ensure: Optimized ABS ratio α.
1: Initialize $Q_\alpha(s, a)$, state s and n=0;
2: Setting learning rate λ_1, greedy probability ε_1, discount factor γ_1 and $threshold_1$;
3: **while** $n <= threshold_1$ **do**
4: In state s, select the optimal action a with greedy probability ε_1;
5: Observe r;
6: randomly transfer from s to $s^{'}$;
7: Update $Q_\alpha(s, a)$ according to Formula (18);
8: $s \leftarrow s^{'}$;
9: $n = n + 1$;
10: **end while**
11: Output: $\alpha = a$;

5.2.2. Sub-Problem B: Given the Abs Ratio α and the Number of Sbs Activation $|S|$ for Optimizing the Cre Bias β

The action of the sub-problem B is $a_t = (\alpha, \beta, |S|)$ where α and $|S|$ are known. $\beta \in$ B and the action space size is $|B|$. State $s = s_t$. Like Formula (18), the updating rule of sub-Q-value can be written as

$$Q_\beta(s, a) \leftarrow Q_\beta(s, a) + \lambda_2 \left[r + \gamma_2 \max_{a'} Q_\beta \left(s^{'}, a^{'} \right) - Q_\beta(s, a) \right] \tag{19}$$

and shown in Algorithm 3.

Algorithm 3: The ABS ratio α and the number of SBSs activation $|S|$ are given to optimize the CRE bias β.

Require: $a_t = (\alpha, \beta, |S|)$.
Ensure: Optimized CRE bias β.
1: Initialize $Q_\beta(s, a)$, state s and n=0;
2: Setting learning rate λ_2, greedy probability ε_2, discount factor γ_2, and $threshold_2$;
3: **while** $n <= threshold_2$ **do**
4: In state s, select the optimal action a with greedy probability ε_2;
5: Observe r;
6: randomly transfer from s to $s^{'}$;
7: Update $Q_\beta(s, a)$ according to Formula (19) ;
8: $s \leftarrow s^{'}$;
9: $n = n + 1$;
10: **end while**
11: Output: $\beta = a$;

5.2.3. Sub-Problem C: Given the Abs Ratio α and the Cre Bias β for Optimizing the Number of Sbs Activation $|S|$

The action of the sub-problem B is $a_t = (\alpha, \beta, |S|)$ where α and β are known. $|S| \in \{0, 1, ..., |P|\}$ and the action space size is $|P| + 1$. State $s = s_t$. It is similar to Formula (18) and the updating rule of sub-Q-value can be written as

$$Q_{|S|}(s, a) \leftarrow Q_{|S|}(s, a) + \lambda_3 \left[r + \gamma_3 \max_{a'} Q_{|S|} \left(s^{'}, a^{'} \right) - Q_{|S|}(s, a) \right] \tag{20}$$

and shown in Algorithm 4.

Algorithm 4: The ABS ratio α and the CRE bias β are given to optimize the number of SBS activation $|S|$.

Require: $a_t = (\alpha, \beta, |S|)$.
Ensure: Optimized ABS ratio α.
1: Initialize $Q_{|S|}(s, a)$, state s and n=0;
2: Setting learning rate λ_3, greedy probability ε_3, discount factor γ_3, and $threshold_3$
3: **while** $n <= threshold_3$ **do**
4: In state s, select the optimal action a with greedy probability ε_3;
5: Observe r;
6: randomly transfer from s to $s^{'}$;
7: Update $Q_{|S|}(s, a)$ according to Formula (20);
8: $s \leftarrow s^{'}$;
9: $n = n + 1$;
10: **end while**
11: Output: $|S| = a$;

The TQL algorithm solves the original problem (11) shown in Algorithm 5.

Algorithm 5: The algorithm for optimizing initial problems.

Require: $\alpha \in$ A, $\beta \in$ B , $|S| = \{0, 1, ..., |S|_{\max}\}$, Reward r, Learning rate $\lambda = \{\lambda_1, \lambda_2, \lambda_3\}$, Greedy probability $\varepsilon = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ and Discount factor $\gamma = \{\gamma_1, \gamma_2, \gamma_3\}$.
Ensure: Optimal action configuration $\{\alpha, \beta, |S|\}$ in each state .
1: Initialize U_t, α, β, $|S|$,
2: **while** $n <= threshold_4$ **do**
3: Fixed the CRE bias β and the number of SBS activation $|S|$, calculate the ABS ratio α according to Algorithm 2. Pass the solved α to step (4) and step (5);
4: Fixing the ABS ratio α and the number of SBS activation $|S|$, calculate the CRE bias β according to Algorithm 3. Pass the solved β to step (3) and step (4);
5: Fix the ABS ratio α and the CRE bias β, calculate the number of SBS activation $|S|$ according to Algorithm 4. Pass the solved $|S|$ to step (4) and step (3);
6: $n = n + 1$;
7: **end while**
8: Output: α, β, $|S|$;

In summary, our scheme has changed the size of action space from traditional exponential increase as $X = (|P| + 1) \times |\mathrm{A}| \times |\mathrm{B}|$ to linear increase as $X = (|P| + 1) + |\mathrm{A}| + |\mathrm{B}|$, which greatly reduces the dimension and size of the action space.

Algorithm 5 can be considered as a multi-stage decision process optimization problem which is shown in Figure 2. The action spaces of the third, fourth, and fifth step in Algorithm 5, which are the optimization problem of Algorithms 2–4, are set to $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$ where $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$ are limited and denoted as set A, B, and $\{0, 1, ..., |P|\}$, respectively. It can be seen in Figure 2 that the state spaces of the third, fourth, and fifth step in Algorithm 5 are the Cartesian product of the other two action spaces and the state spaces size are $|\mathscr{B}||\mathscr{C}|$, $|\mathscr{A}||\mathscr{C}|$ and $|\mathscr{A}||\mathscr{B}|$, respectively. The state transition equation refers to the transition probability from state $b_i c_i$ to state $a_i b_i$ conditioned on taking action a_i. The state transition probability is written as $P\left(s^{'} = a_i b_i | s = b_i c_i, action = a_i\right) = 1$, where $a_i \in \mathscr{A}$, $b_i \in \mathscr{B}$ and $c_i \in \mathscr{C}$. We assume with the condition that there is no interference in the transition process, so the transition probability here is 1. If the multi-stage decision-making process is a closed loop that $b_i c_i = b_k c_k$ or $a_i c_i = a_k c_k$ or $a_i b_j = a_k b_l$ exists, Algorithm 5 is stable. Since $|\mathscr{A}|$, $|\mathscr{B}|$, and $|\mathscr{C}|$ are all bounded, $b_i c_i = b_k c_k$ or $a_i c_i = a_k c_k$ or $a_i b_j = a_k b_l$ appears at most $\min\left(|\mathscr{B}||\mathscr{C}|, |\mathscr{A}||\mathscr{C}|, |\mathscr{A}||\mathscr{B}|\right) + 1$ transitions during the whole stage. Therefore, it indicates that Algorithm 5 is stable. In Section 6, Figure 6 further illustrates that Algorithm 5 converges to a near optimal solution.

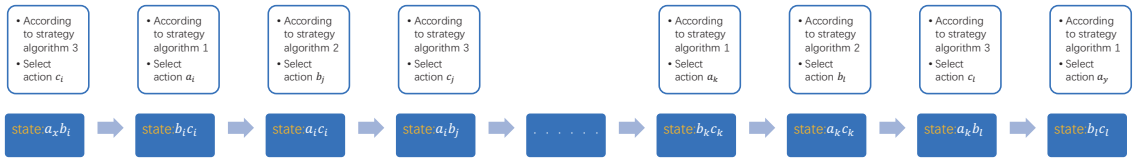

Figure 2. Multi-stage decision process.

5.3. Neural Network for the Classification of States

In subsection B, the TQL algorithm solves the problem of action space dimension explosion. In order to have the ability to classify the state for the agent, a DNN whose structure is shown in Figure 3, in which there are two hidden layers and each hidden layer has 512 nodes. The activation function for each hidden layer is a rectified linear unit (ReLU), and ADAM (adaptive moment estimation) [36] is utilized as updating algorithm and learning-rate is equal to 0.001. The optimal strategy of TQL is to label the samples and specify optimal action in each state. Samples are the set of users position which are gathered from TQL algorithm. The input is the location information of users and the optimal action is encoded according to the index in the action space as the output.

We tried a one-layer, two-layer, and three-layer hidden layer network, and the experimental results showed that it had similar performance on the training performance. Two hidden layers DNN performed relatively more effective with respect to training speed and performance, which is why we use two hidden layers of DNN.

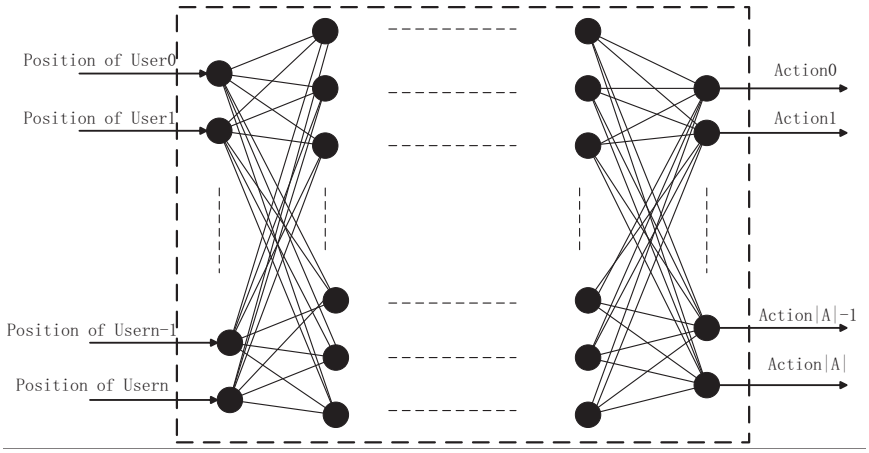

Figure 3. The inputs are the users' position, labels are the optimal action in each state.

6. Numerical Simulation

The parameters of experimental simulation are set according to the 3GPP LTE-A HetNets framework [37], and the wireless channel is modeled as deterministic path loss attenuation and random shadow fading models. In this part, the scenario we deployed is that each MBS covers the users in a 120° cell as shown as shadow part in Figure 4 and is interfered by three other MBSs. We deployed a field where six SBSs are randomly deployed within the coverage of the MBS in the green shaded part of Figure 4 and select working mode according to load conditions.

The coverage radius of the MBS and the SBSs are 500 m and 100 m, respectively. The thermal noise power is −176 dBm, the system spectrum bandwidth is 10 MHz and the antenna gains of the MBSs, SBSs, and the UE are 14 dBi, 5 dBi, and 0 dBi, respectively. The maximum transmission power of the MBSs and the SBSs are set to 46 dBm and 30 dBm, respectively. The probability of a user entering a cell to access a MBS and a SBS are 1/3 and 2/3, respectively. The proportion φ of the BSs that wake up is set here to 0.5. Although LTE frame includes 10 subframes, the ABS mode has a periodicity of eight subframes. The ABS ratio of protected subframe to traditional subframe $\alpha \in [0, 1]$ belongs to the set

$\{0/8, ..., 7/8\}$. CRE is denoted by $\beta \in B$, where B $= \{0, 6, 9, 12, 15, 18\}$. The specific parameters are listed in Table 1.

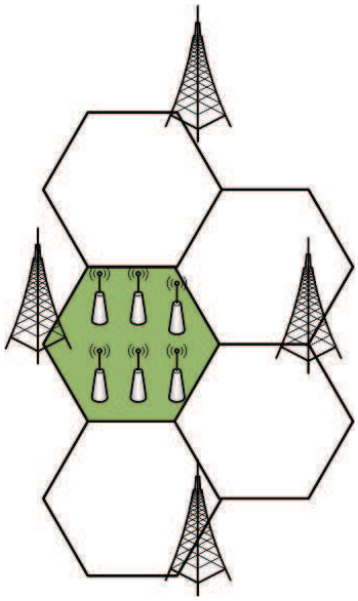

Figure 4. Simulation scenario.

Table 1. Simulation Parameters

Parameter	Value
Macro base station radius	600 m
Small base station radius	100 m
Minimum distance between small base station	40 m
Minimum distance from macro base station to small base station	70 m
System bandwidth B	10 MHz
Transmit power of MBS	Normal: 46 dBm; Sleeping: 12 dBm; $N_{TRX} = 6$
Transmit power of SBS	Normal: 30 dBm; $N_{TRX} = 2$
Noise power density N_0	-174 dBm
Path loss from MBS to UE	128.1 + 37.6 $\times$ $log_{10}(R[\text{km}])$, R is the distance between MBS and UE
Path loss from SBS to UE	149.1 + 37.6 $\times$ $log_{10}(R[\text{km}])$, R is the distance between MBS and UE
Carrier frequency	2.0 Ghz
File size	0.5 mbytes

Figures 5 and 6 show the relationship during iterations and accuracy of Q-Learning algorithm and our TQL algorithm where learning rate, discount factor, and greed rate are all set to 0.1, and the number of users are set to 50, 100, 150, and 200, respectively. Figure 5 shows that the tabular method of the Q-Learning algorithm converges after $80 \times 1000 = 800{,}000$ iterations under different load conditions. Our proposed TQL algorithm converges after $800 \times 400 = 320{,}000$ iterations as shown in Figure 6. We can see that the convergence speed of our proposed TQL algorithm is increased by about 60% compared with the Algorithm 1. Note that the convergence speed of TQL algorithm proposed in this paper is still much faster than that of Algorithm 1, especially in the case where the action space cardinality is very large from the analysis of Algorithm 5.

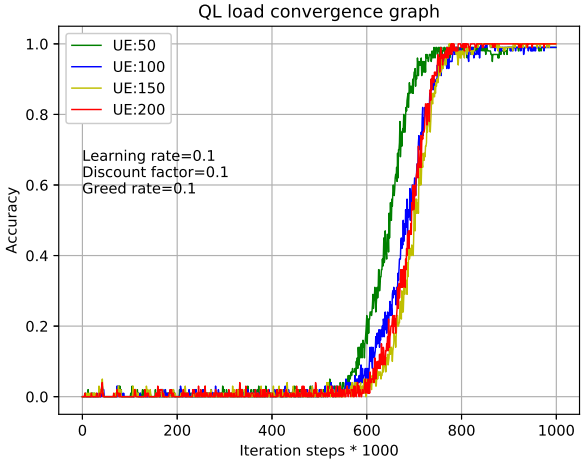

Figure 5. QL algorithm convergence graph with different loads.

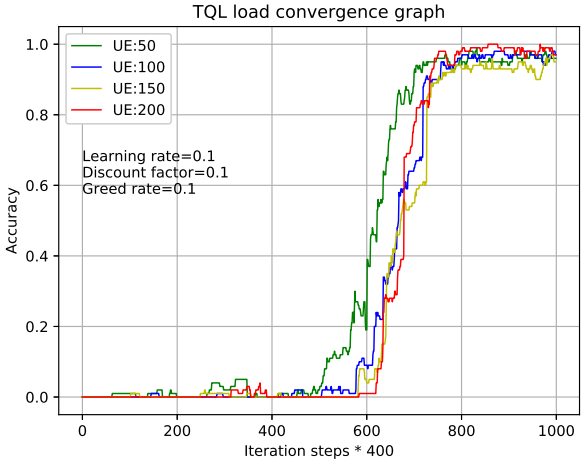

Figure 6. TQL algorithm convergence graph with different loads.

Figure 7 shows that the comparison of sub-QLs (Algorithms 2–4) iterations in the TQL algorithm give different results where relative parameters are the same as that in analysis of Figure 5 and the number of users is 100. Take the greed line and red line as examples which indicate the number of sub-QL iterations of SBSs ($threshold_3$ in Algorithm 4), ABS ($threshold_1$ in Algorithm 2), and CRE ($threshold_2$ in Algorithm 3) are all set to 400 and 2000, respectively. When the iteration number of TQL represented by the green line and red line is more than about 75 and 30, our proposed TQL algorithm is convergent, but the final accuracy rates are about 98% and 90%, respectively. We can see that, although the convergence speed respected by green line is relatively slower, the final correct rate is higher and the performance is better than that respected by the red line. Our proposed TQL algorithm can make a balance between performance and convergence speed according to actual requirements by adjusting the iterations of the subsystems such as the black line indicated case where the numbers of sub-QL iterations of SBSs, ABS, and CRE are set to 1000, 500, and 500, respectively. The number of iterations for TQL represented by cyan line is more than about 50 and the final correct rate is about 93%.

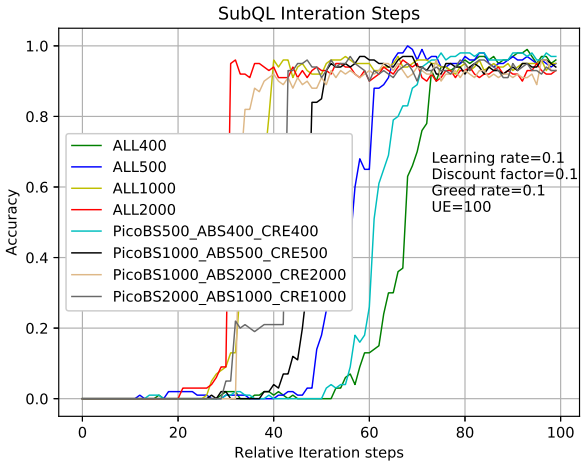

Figure 7. TQL with different sub-QL iteration numbers convergence graph.

In the case where other experimental parameters are the same as that of Figure 7, Figure 8 shows the influence of different learning rates of sub-QL algorithms on convergence speed and correct rate. Take the red line and the cyan line as examples which indicate that the learning rates of sub-QL of SBSs, ABS, and CRE are set to 0.1, 0.05, 0.05 and 0.05, 0.01, 0.01, respectively. We can see that, when the iteration numbers of TQL respected by the red line and cyan line are about 30 and 70, our TQL algorithm is convergent and the final accuracy rate are about 95% and 99%, respectively. The convergence speed respected by the red line is faster than that respected by the cyan line, but the final accuracy rate is otherwise. Pay attention to the problem caused by the setting learning rates respected by the green line which are all set to 0.01, if the learning rate is set too low, the system falls into the local optimum, and the global optimum cannot be found. It is easy to make this mistake in the RL.

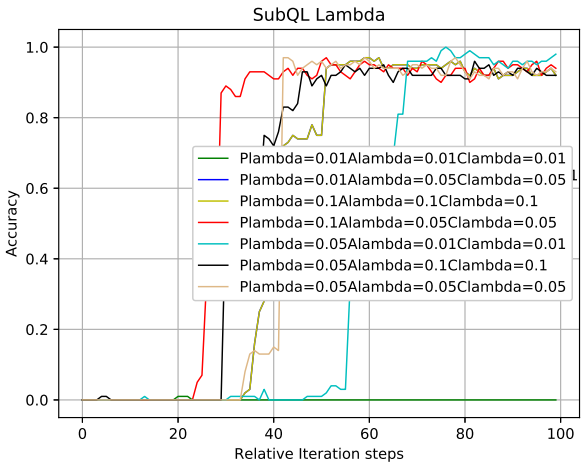

Figure 8. TQL algorithms convergence graph with different learning rates.

In the case where other experimental parameters the same as that in analysis of Figures 7 and 8, the methods of analyzing Figures 9 and 10 are like that in Figures 7 and 8. The results obtained from Figures 7–10 are that the balance can be obtained between performance and convergence speed by changing the corresponding parameters of the sub-QL in our TQL algorithm. It can be seen that our TQL algorithm has greater flexibility in parameter adjustment compared to the general system where only one set of parameters is set.

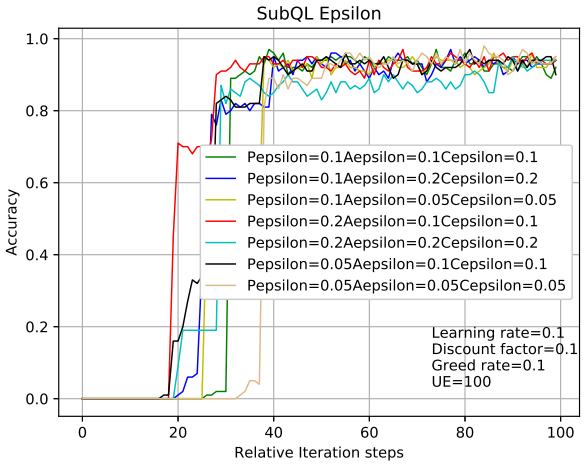

Figure 9. TQL algorithms convergence graph with different greedy rates.

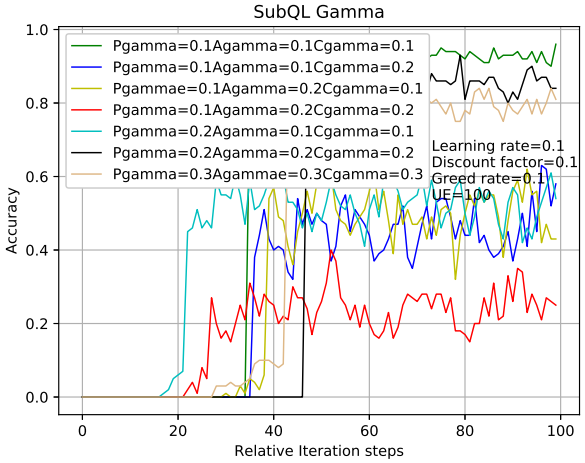

Figure 10. TQL algorithms' convergence graph with different discount factors.

Figures 11–13 show examples of the sample classification of our designed DNN when the ABS, CRE, and the number of SBSs activations are set to (0, 0, 6), (3/8, 6, 6) and (7/8, 18, 6), respectively. The labeled samples are obtained from the optimal strategy of TQL algorithm, in which 90% of them are training samples and the rest are test samples. The red dots represent the macro base stations where the macro base station with the number 0 is the cell signal source, and the macro base stations with the other numbers are the interference signal sources. Blue and green dots indicate the location of SBSs and users in the cell, respectively.

Figure 14 shows that QL, TQL, and the ADP ES IC algorithm in [9] optimize the power consumption of HetNets. We can see that, under the condition that the number of users being less than 100, the power consumption obtained by the QL algorithm is lower than the TQL proposed in this paper. However, when the number of users is greater than 100, the power consumptions of the two algorithms are the same, which are between the maximum power consumption and the minimum power consumption. The algorithm of ADP ES IC optimizes the power consumption best in the case of 50 users, and it is seen in Figure 14 that the power consumption control of the entire system is better than QL and TQL in the entire 10–200 users interval. However, it can be seen that such good results is obtained at the premise of sacrificing energy efficiency in Figure 15.

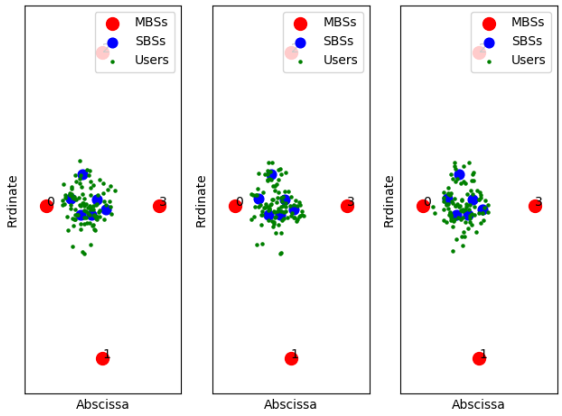

Figure 11. ABS = 0, CRE = 0.

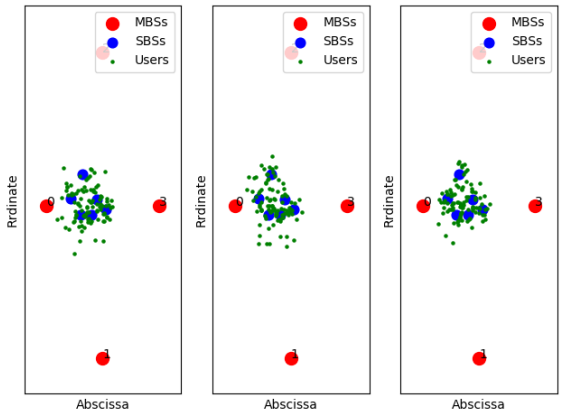

Figure 12. ABS = 3/8, CRE = 6.

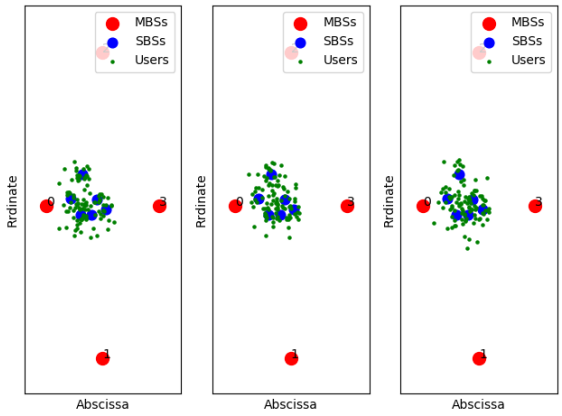

Figure 13. ABS = 7/8, CRE = 18.

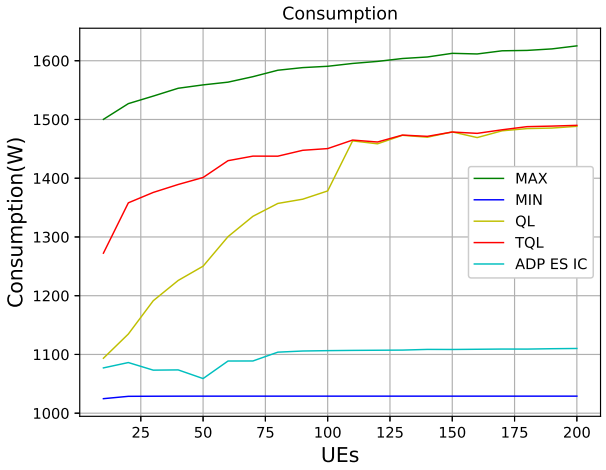

Figure 14. The consumption graphs of different algorithms.

Figure 15 shows the energy efficiency of HetNets is optimized by QL, TQL, and ADP ES IC algorithms, respectively. Green solid line and red dotted line represent the theoretical optimal energy efficiency and the energy efficiency obtained by our TQL algorithm, respectively. The sub-picture on the left of Figure 15 shows that the optimized energy efficiency of our TQL algorithm is very close to the theoretical optimal, and the sub-picture on the right of Figure 15 shows that the index of energy efficiency of our algorithm optimization system is slightly lower than the theoretical optimal. According to the analysis of Figure 6, because the TQL algorithm has not found the optimal solutions (i.e., Pico BS, ABS, and CRE configuration) in some states, the gap exists between theoretical optimal and TQL algorithm, as shown in the sub-picture on the right of Figure 15. However, Figure 15 proves that the TQL energy efficiency performance is very close to the theoretical optimal energy efficiency, indicating that the TQL algorithm proposed in this paper has not been optimized in some states, but the solution found is also a relatively optimal solution, which may be a suboptimal solution. For the system, the performance loss is small. The ADP ES IC algorithm is poor in energy efficiency optimization, mainly because the authors focus on power optimization of the system in HetNets.

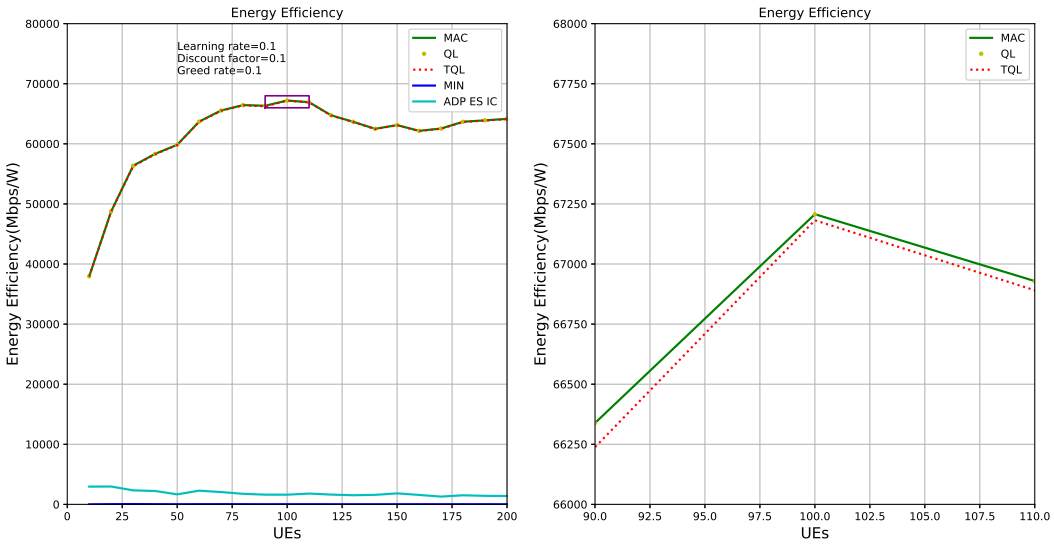

Figure 15. Energy efficiency optimization graphs of different algorithms.

7. Conclusions

In order to jointly optimize resources to maximize the energy efficiency of HetNets by RL, there is a problem of too large action space and state space. We propose a novel QL algorithm (TQL) based on the multi-stage decision process to improve the QL algorithm to effectively solve the problem of excessive action space. Through the analysis of Algorithm 5, the advantage is greater in the case of large action space. For the dimension problem of state space, DNN is designed to classify states where the optimization policy of novel Q-Learning is set to label samples. At the same time, compared with the general RL algorithm, there is only one set of adjustable parameters, and the TQL learning proposed in this paper can further adjust the parameters according to the system requirements to further optimize the system. Thus far, the dimensions of action and state have been solved. The algorithm proposed in this paper is more flexible. Finally, the simulation result proves that the algorithm proposed in this paper is effective and feasible, and improves the convergence speed by 60% compared with the tabular QL.

Author Contributions: Formal analysis, Z.L.; Methodology, X.W. and J.L.; Project administration, J.L.; Software, L.L. and J.L.; Writing—original draft, L.L.; Writing—review & editing, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Zhejiang Provincial Natural Science Foundation of China under Grant No. Y20F010069, as well as supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 51874264.

Acknowledgments: This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. Y20F010069, as well as supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 51874264 and Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China. (Corresponding author: Jun Li.)

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shafique, K.; Khawaja, B.A.; Sabir, F.; Qazi, S.; Mustaqim, M. Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access* **2020**, *8*, 23022–23040. [CrossRef]
2. Baidas, M.W.; Al-Mubarak, M.; Alsusa, E.; Awad, M.K. Joint Subcarrier Assignment and Global Energy-Efficient Power Allocation for Energy-Harvesting Two-Tier Downlink NOMA Hetnets. *IEEE Access* **2019**, *7*, 163556–163577. [CrossRef]
3. Chen, Y.; Zhang, N.; Zhang, Y.; Chen, X.; Wu, W.; Shen, X.S. TOFFEE: Task Offloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing. *IEEE Trans. Cloud Comput.* **2019**, 1. [CrossRef]
4. Wei, Y.; Yu, F.R.; Song, M.; Han, Z. User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach. *IEEE Wirel Commun.* **2017**, *17*, 680–692. [CrossRef]
5. Asuhaimi, F.A.; Bu, S.; Klaine, P.V.; Imran, M.A. Channel access and power control for energy-efficient delay-aware heterogeneous cellular networks for smart grid communications using deep reinforcement learning. *IEEE Access* **2019**, *7*, 133474–133484. [CrossRef]
6. Amiri, R.; Almasi, M.A.; Andrews, J.G.; Mehrpouyan, H. Reinforcement learning for self organization and power control of two-tier heterogeneous networks. *IEEE Wirel. Commun.* **2019**, *18*, 3933–3947. [CrossRef]
7. Ye, J.; Zhang, Y.J. DRAG: Deep reinforcement learning based base station activation in heterogeneous networks. *IEEE Mob. Comput.* **2019**, *19*, 2076–2087. [CrossRef]
8. Ayala-Romero, J.A.; Alcaraz, J.J.; Vales-Alonso, J. Data-driven configuration of interference coordination parameters in HetNets. *IEEE Veh. Technol.* **2018**, *67*, 5174–5187. [CrossRef]
9. Ayala-Romero, J.A.; Alcaraz, J.J.; Vales-Alonso, J. Energy saving and interference coordination in HetNets using dynamic programming and CEC. *IEEE Access* **2018**, *6*, 71110–71121. [CrossRef]
10. Nasser, A.; Muta, O.; Elsabrouty, M.; Gacanin, H. Compressive Sensing Based Spectrum Allocation and Power Control for NOMA HetNets. *IEEE Access* **2019**, *7*, 98495–98506. [CrossRef]

11. Chang, K.C.; Chu, K.C.; Wang, H.C.; Lin, Y.C.; Pan, J.S. Energy Saving Technology of 5G Base Station Based on Internet of Things Collaborative Control. *IEEE Access* **2020**, *8*, 32935–32946. [CrossRef]
12. Li, J.; Wang, X.; Li, Z.; Wang, H.; Li, L. Energy Efficiency Optimization Based on eICIC for Wireless Heterogeneous Networks. *IEEE Internet Things* **2019**, *6*, 10166–10176. [CrossRef]
13. Yin, F.; Wang, A.; Liu, D.; Zhang, Z. Energy-aware joint user association and resource allocation for coded cache-enabled hetnets. *IEEE Access* **2019**, *7*, 94128–94142. [CrossRef]
14. Anany, M.; Elmesalawy, M.M.; El-Haleem, A.M.A. Matching Game-Based Cell Association in Multi-RAT HetNet Considering Device Requirements. *IEEE Internet Things* **2019**, *6*, 9774–9782. [CrossRef]
15. Wang, Y.; Yang, H.H.; Zhu, Q.; Quek, T.Q. Analysis of Packet Throughput in Spatiotemporal HetNets with Scheduling and Various Traffic Loads. *IEEE Wirel. Commun.* **2019**, *9*, 95–98. [CrossRef]
16. Dong, X.; Zheng, F.C.; Zhu, X.; OFarrell, T. On the Local Delay and Energy Efficiency of Clustered HetNets. *IEEE Veh. Technol.* **2019**, *68*, 2987–2999. [CrossRef]
17. Khan, S.A.; Kavak, A.; Çolak, S.A.; Küçük, K. A Novel Fractional Frequency Reuse Scheme for Interference Management in LTE-A HetNets. *IEEE Access* **2019**, *7*, 109662–109672. [CrossRef]
18. Tiwari, R.; Deshmukh, S. MVU Estimate of User Velocity via Gamma Distributed Handover Count in HetNets. *IEEE Commun. Lett.* **2019**, *23*, 482–485. [CrossRef]
19. Tiwari, R.; Deshmukh, S. Prior Information-Based Bayesian MMSE Estimation of Velocity in HetNets. *IEEE Wirel. Commun.* **2018**, *8*, 81–84. [CrossRef]
20. Chen, N.; Zhang, X.; Sun, S. An Adaptive Coverage Enhancement Scheme Based on mmWave RoF for Future HetNets. *IEEE Access* **2019**, *7*, 29107–29113. [CrossRef]
21. Deb, S.; Monogioudis, P. Learning-based uplink interference management in 4G LTE cellular systems. *IEEE ACM Netw.* **2014**, *23*, 398–411. [CrossRef]
22. Chen, L.; Ma, L.; Xu, Y.; Leung, V.C. Hypergraph spectral clustering based spectrum resource allocation for dense NOMA-HetNet. *IEEE Wirel. Commun.* **2018**, *8*, 305–308. [CrossRef]
23. Siddavaatam, R.; Woungang, I.; Anpalagan, A. Joint optimisation of radio and infrastructure resources for energy-efficient massive data storage in the mobile cloud over 5G HetNet. *IET Wirel. Sens. Syst.* **2019**, *9*, 323–332. [CrossRef]
24. Huang, X.; Xu, W.; Xie, G.; Jin, S.; You, X. Learning oriented cross-entropy approach to user association in load-balanced HetNet. *IEEE Wirel. Commun.* **2018**, *7*, 1014–1017. [CrossRef]
25. Castro-Hernandez, D.; Paranjape, R. Classification of user trajectories in LTE HetNets using unsupervised shapelets and multiresolution wavelet decomposition. *IEEE Veh. Technol.* **2017**, *66*, 7934–7946. [CrossRef]
26. Castro-Hernandez, D.; Paranjape, R. Optimization of handover parameters for LTE/LTE-A in-building systems. *IEEE Veh. Technol.* **2017**, *67*, 5260–5273. [CrossRef]
27. Sun, Y.; Feng, G.; Qin, S.; Liang, Y.; Yum, T.P. The SMART Handoff Policy for Millimeter Wave Heterogeneous Cellular Networks. *IEEE Mob. Comput.* **2018**, *17*, 1456–1468. [CrossRef]
28. Fan, Y.; Zhang, Z.; Li, H. Message Passing Based Distributed Learning for Joint Resource Allocation in Millimeter Wave Heterogeneous Networks. *IEEE Wirel. Commun.* **2019**, *18*, 2872–2885. [CrossRef]
29. Li, W.; Wang, J.; Li, L.; Zhang, G.; Dang, Z.; Li, S. Intelligent Anti-Jamming Communication with Continuous Action Decision for Ultra-Dense Network. In Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
30. Chai, R.; Cheng, J.; Pu, X.; Chen, Q. Neural network based vertical handoff performance enhancement in heterogeneous wireless networks. In Proceedings of the 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 23–25 September 2011; pp. 1–4.
31. Fan, S.; Tian, H.; Sengul, C. Self-optimization of coverage and capacity based on a fuzzy neural network with cooperative reinforcement learning. *EURASIP J. Wirel. Commun. Netw.* **2014**, *2014*, 57. [CrossRef]
32. Lei, L.; You, L.; Dai, G.; Vu, T.X.; Yuan, D.; Chatzinotas, S. A deep learning approach for optimizing content delivering in cache-enabled HetNet. In Proceedings of the 2017 International Symposium on Wireless Communication Systems (ISWCS), Bologna, Italy, 28–31 August 2017; pp. 449–453.
33. Sun, H.; Lv, T.; Zhang, X.; Liu, Z. Convolution neural network based dynamic pico cell operation for multi-tier heterogeneous networks. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–6.
34. ETSI TS 136 300 V10. 4.0. *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN): Overall Description*; ETSI: Sophia Antipolis, France, 2011.

35. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*; MIT Press: Cambridge, UK, 1998.
36. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
37. E-UTRA. *Further Advancements for E-UTRA Physical Layer Aspects*; 3GPP TS 36.814; V9. 0.0; ETSI: Sophia Antipolis, France, 2010.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).